



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS QUIXADÁ**  
**CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**PEDRO RODRIGUES DE CARVALHO JÚNIOR**

**UM CATÁLOGO DE EXTENSÕES DA LINGUAGEM DE MODELAGEM *KAOS***

**QUIXADÁ**

**2022**

PEDRO RODRIGUES DE CARVALHO JÚNIOR

UM CATÁLOGO DE EXTENSÕES DA LINGUAGEM DE MODELAGEM *KAOS*

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus Quixadáda Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Orientador: Prof. Dr. Enyo José Tavares Gonçalves

QUIXADÁ

2022

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)

---

C327c Carvalho Júnior, Pedro Rodrigues de.  
Um catálogo de extensões da linguagem de modelagem Kaos / Pedro Rodrigues de Carvalho Júnior. –  
2022.  
80 f.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá,  
Curso de Ciência da Computação, Quixadá, 2022.  
Orientação: Prof. Dr. Enyo José Tavares Gonçalves.

1. Engenharia de Requisitos. 2. UML (Computação). 3. Catálogos. I. Título.

CDD 004

---

PEDRO RODRIGUES DE CARVALHO JÚNIOR

UM CATÁLOGO DE EXTENSÕES DA LINGUAGEM DE MODELAGEM *KAOS*

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação do Campus Quixadáda Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da Computação.

Aprovada em: \_\_\_\_/\_\_\_\_/\_\_\_\_.

BANCA EXAMINADORA

---

Prof. Dr. Enyo José Tavares Gonçalves (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Marcos Antonio de Oliveira  
Universidade Federal do Ceará - Campus Quixadá  
(UFC)

---

Profa. Dra. Rainara Maia Carvalho  
Universidade Federal do Ceará - Campus Quixadá  
(UFC)

Dedico esse trabalho ao meu pai (*in memoriam*),  
minha mãe e meus irmãos.

## **AGRADECIMENTOS**

Quero agradecer primeiro a minha mãe pelo amor incondicional, incentivo e suporte para que eu pudesse chegar até aqui. Quero agradecer também aos meus irmãos pelo apoio e incentivo nessa jornada.

Quero agradecer a Lola, Hortência, Ernilton, Adriano que foram as pessoas que me acolheram em Quixadá e garantiram a minha permanência ali me ofertando moradia de graça durante todo o período da graduação.

Quero agradecer aos meus amigos e destacar aqui o nome de Bruno que nos fins de semana quando eu queria vir para São José se dispunham a viajar 30 km de moto para me buscar na sexta a noite e me deixar na segunda de madrugada para pegar o ônibus.

Quero agradecer também Prof. Dr. Enyo José Tavares Gonçalves pelo excelente trabalho de orientação nessa difícil tarefa que é o Trabalho de Conclusão de Curso.

Quero agradecer também a todos os professores que contribuíram para minha formação, assim como todos os funcionários da instituição que faziam ela funcionar da forma eficiente como funciona.

E por fim agradecer aos meus colegas e amigos que ganhei durante o curso e que fizeram os meus dias um pouco mais felizes.

“E você aprende que realmente pode suportar. . .  
que realmente é forte, e que pode ir muito mais  
longe depois de pensar que não se pode mais. E  
que realmente a vida tem valor e que você tem  
valor diante da vida! Nossas dúvidas são traido-  
ras e nos fazem perder o bem que poderíamos  
conquistar se não fosse o medo de tentar.”

(William Shakespeare)

## RESUMO

*Knowledge Acquisition Automated Specification* (KAOS) é uma linguagem de modelagem cujo foco principal é representar explicitamente os objetivos dos sistemas, os conflitos e obstáculos entre esses objetivos, quais os objetos responsáveis por satisfazê-los e quais as consequências do resultado da interação entre os mesmos. A linguagem de modelagem KAOS vem sendo estendida para diversos domínios/áreas de aplicação como segurança de sistemas, sistemas adaptativos e aspectos. Identificar uma extensão de KAOS ou seus construtores com base em suas características, como a área de aplicação, por exemplo, pode ser uma tarefa complexa que requer grande quantidade de tempo e tem uma propensão a não ser exitosa quando executada sem o apoio adequado. Notamos então a necessidade de criar um catálogo para armazenar as extensões da linguagem de modelagem KAOS, de modo a contribuir com a identificação destas extensões e seus construtores. O catálogo proposto neste trabalho tem como objetivo facilitar a identificação de extensões de KAOS e seus construtores com base em suas características, como a área de aplicação ou tipo de construtor. Ele é baseado nos resultados de uma Revisão Sistemática da Literatura (RSL) executada em 2019 e atualizada em 2021 que identificou e analisou as extensões de KAOS existentes e os resultados apontam para 42 extensões identificadas e 183 construtores propostos. O catálogo foi ainda avaliado por meio de um estudo exploratório baseado em *survey* que teve como objetivo analisar qual o efeito do uso do catálogo na identificação das extensões de KAOS e seus construtores. Na avaliação um questionário foi proposto para ser respondido pelos participantes do estudo, onde os mesmos teriam que usar o catálogo para obter as respostas do questionário com perguntas relacionadas a identificação das extensões e seus construtores. Como resultado temos o catálogo disponível no link <https://kaos-catalogue-d9e49.web.app/>. Os resultados da avaliação mostram que pelo menos 90% dos participantes obteve sucesso ao responder pelo menos 12 (doze) das 15 (quinze) perguntas propostas e ainda que em aproximadamente 75% das questões os participantes julgaram como muito fácil ou fácil obter a resposta usando o catálogo. Finalmente, esse trabalho apresenta discussões e trabalhos futuros.

**Palavras-chave:** Engenharia de Requisitos. UML (Computação). Catálogos



## ABSTRACT

Knowledge Acquisition Automated Specification (KAOS) is a modeling language whose main focus is to explicitly represent the goals of systems, the conflicts and obstacles between these goals, which objects are responsible for satisfying them, and the consequences of the outcome of the interaction between them. The KAOS modeling language has been extended to various domains/application areas such as system security, adaptive systems, and aspects. Identifying a KAOS extension or its constructs based on its characteristics, such as application area, for example, can be a complex task that requires a large amount of time and has a propensity to be unsuccessful when performed without proper support. We then noticed the need to create a catalog to store the extensions of the KAOS modeling language, in order to contribute to the identification of these extensions and their constructors. The catalog proposed in this paper aims to facilitate the identification of KAOS extensions and their constructors based on their characteristics, such as application area or type of constructor. It is based on the results of a Systematic Literature Review (SLR) performed in 2019 and updated in 2021 that identified and analyzed existing KAOS extensions and the results point to 42 identified extensions and 183 proposed builders. The catalog was further evaluated through an exploratory study based on *survey* that aimed to analyze what effect the use of the catalog has on the identification of KAOS extensions and their builders. In the evaluation a questionnaire was proposed to be answered by the participants of the study, where they would have to use the catalog to obtain the answers to the questionnaire with questions related to the identification of the extensions and their builders. As a result we have the catalog available at the link <https://kaos-catalogue-d9e49.web.app/>. The evaluation results show that at least 90% of the participants succeeded in answering at least twelve (12) of the fifteen (15) proposed questions, and that in approximately 75% of the questions the participants found it very easy or easy to get the answer using the catalog. Finally, this paper presents discussions and future work.

**Keywords:** Requirements Engineering. UML (Computing). Catalogs

## LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas da Engenharia de Requisitos . . . . .	18
Figura 2 – Exemplo do modelo de responsabilidade KAOS. . . . .	21
Figura 3 – Modelo de táticas de <i>deception</i> para autenticação. . . . .	23
Figura 4 – Metodologia usada para criação do catálogo de extensões do KAOS . . . . .	28
Figura 5 – Página Inicial . . . . .	34
Figura 6 – Página de Extensões . . . . .	35
Figura 7 – Página de detalhamento de uma extensão . . . . .	35
Figura 8 – Página de Report . . . . .	36
Figura 9 – Extensões Reportadas . . . . .	36
Figura 10 – Formulário de Cadastro de Extensões . . . . .	37
Figura 11 – Exemplo de fluxo a ser testado . . . . .	37
Figura 12 – Reportar uma nova extensão . . . . .	38
Figura 13 – Extensões Reportadas . . . . .	38
Figura 14 – Exemplo de funcionalidade a ser testada . . . . .	39
Figura 15 – Resultado ao filtrar as extensões pela área de aplicação Riscos . . . . .	39
Figura 16 – Exemplo de tarefa do questionário . . . . .	42
Figura 17 – Curso e semestre . . . . .	43
Figura 18 – Conhecimentos prévios . . . . .	44
Figura 19 – Consultar Extensão . . . . .	45
Figura 20 – Extensões por área de aplicação específica . . . . .	46
Figura 21 – Quantidade de publicações por autor específico . . . . .	47
Figura 22 – Quantidade de publicações por ano específico . . . . .	48
Figura 23 – Apresenta definição dos conceitos dos construtores . . . . .	49
Figura 24 – Nível de sintaxe de uma extensão específica . . . . .	50
Figura 25 – Tipo de local de uma publicação de uma extensão . . . . .	51
Figura 26 – Local de publicação de uma extensão . . . . .	52
Figura 27 – Extensão possui ferramenta suporte . . . . .	53
Figura 28 – Forma de validação de uma extensão . . . . .	54
Figura 29 – Quantidade de construtores propostos em uma extensão . . . . .	55
Figura 30 – Construtor pertence há uma extensão . . . . .	56
Figura 31 – Qual o tipo de um construtor específico . . . . .	57

Figura 32 – Tipo de construtor mais proposto . . . . .	58
Figura 33 – Extensão relacionada a um construtor específico . . . . .	59
Figura 34 – Opinião dos participantes experiencia de utilizar o catálogo . . . . .	60
Figura 35 – Impressões sobre layout do catálogo . . . . .	61
Figura 36 – Impressões sobre a complexidade das tarefas . . . . .	61
Figura 37 – Impressões sobre velocidade e corretude das consultas . . . . .	61
Figura 38 – Console do Realtime Database . . . . .	79
Figura 39 – Exemplo de lista dos construtores dentro do JSON de uma extensão . . . . .	80
Figura 40 – Console do Cloud Storage . . . . .	80

## LISTA DE TABELAS

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto. . . . .	27
Tabela 2 – Informações extraídas das extensões . . . . .	30
Tabela 3 – Informações extraídas dos Construtores . . . . .	31
Tabela 4 – Tecnologias utilizadas . . . . .	31

## LISTA DE ABREVIATURAS E SIGLAS

CRS	<i>Catálogo de Requisitos de Software</i>
DSML	<i>Domain-Specific Modeling Languages</i>
GPML	<i>General-Purpose Modeling Languages</i>
HTML	<i>HyperText Markup Language</i>
JSON	<i>JavaScript Object Notation</i>
KAOS	<i>Knowledge Acquisition Automated Specification</i>
MBE	<i>Model-Based Engineering</i>
NoSQL	<i>Not Only Structured Query Language</i>
PCOMP	<i>Programa de Pós-Graduação em Computação</i>
RNFs	<i>Requisitos Não-Funcionais</i>
RSL	<i>Revisão Sistemática da Literatura</i>
SDKs	<i>Software Development kits</i>
SIG	<i>Softgoal Interdependency Graph</i>
SQL	<i>Structured Query Language</i>
UI	<i>User Interface</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>15</b>
<b>1.1</b>	<b>Objetivos</b>	<b>16</b>
<b>1.1.1</b>	<i>Objetivo Geral</i>	<b>16</b>
<b>1.1.2</b>	<i>Objetivos específicos</i>	<b>16</b>
<b>1.2</b>	<b>Organização do trabalho</b>	<b>17</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>18</b>
<b>2.1</b>	<b>Engenharia de Requisitos</b>	<b>18</b>
<b>2.2</b>	<b>Fundamentos sobre Linguagens de modelagem e extensões</b>	<b>19</b>
<b>2.3</b>	<b>KAOS</b>	<b>20</b>
<b>2.4</b>	<b>Extensões de KAOS</b>	<b>21</b>
<b>2.5</b>	<b>Catálogos na Engenharia de Software</b>	<b>23</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>25</b>
<b>4</b>	<b>METODOLOGIA</b>	<b>28</b>
<b>4.1</b>	<b>Extração das Extensões</b>	<b>28</b>
<b>4.2</b>	<b>Desenvolvimento do catálogo</b>	<b>28</b>
<b>4.3</b>	<b>População do Catálogo</b>	<b>29</b>
<b>4.4</b>	<b>Teste Funcional do Catálogo</b>	<b>29</b>
<b>4.5</b>	<b>Avaliação do Catálogo</b>	<b>29</b>
<b>5</b>	<b>CATÁLOGO DE EXTENSÕES DA LINGUAGEM DE MODELAGEM</b>	
	<b>KAOS</b>	<b>30</b>
<b>5.1</b>	<b>Extração das Extensões</b>	<b>30</b>
<b>5.2</b>	<b>Tecnologias</b>	<b>31</b>
<b>5.3</b>	<b>O Catálogo</b>	<b>33</b>
<b>5.3.1</b>	<i>Visão Geral</i>	<b>33</b>
<b>5.4</b>	<b>Apresentação do Catálogo</b>	<b>34</b>
<b>5.5</b>	<b>Teste Funcional do Catálogo</b>	<b>37</b>
<b>5.6</b>	<b>Limitações</b>	<b>39</b>
<b>6</b>	<b>AVALIAÇÃO</b>	<b>41</b>
<b>6.1</b>	<b>Metodologia da Avaliação</b>	<b>41</b>
<b>6.2</b>	<b>Caracterização do Perfil dos Participantes</b>	<b>42</b>

<b>6.3</b>	<b>Resultados da Avaliação</b> . . . . .	44
<b>6.4</b>	<b>Discussões</b> . . . . .	62
<b>6.5</b>	<b>Ameaças a validade</b> . . . . .	63
<b>6.5.1</b>	<i>Validade de face</i> . . . . .	63
<b>6.5.2</b>	<i>Validade de conteúdo</i> . . . . .	63
<b>6.5.3</b>	<i>Validade de conclusão</i> . . . . .	63
<b>6.5.4</b>	<i>Validade de construção</i> . . . . .	64
<b>7</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b> . . . . .	65
	<b>REFERÊNCIAS</b> . . . . .	67
	<b>APÊNDICE A – QUESTIONÁRIO DE AVALIAÇÃO DO CATÁLOGO</b>	69
	<b>APÊNDICE B – TECNOLOGIAS UTILIZADAS</b> . . . . .	79

## 1 INTRODUÇÃO

Em toda sua abrangência, o mundo moderno depende de sistemas computacionais (SOMMERVILLE, 2011). Estes estão presentes nos mais variados setores e domínios como, por exemplo, transportes, saúde, agricultura, educação, entre outros e fornecem suporte à automatização de tarefas.

A engenharia de software surgiu a partir da necessidade de desenvolvimento de sistemas complexos ao proporcionar metodologias e boas práticas na produção de software (SOMMERVILLE, 2011). As metodologias definem, de forma geral, cinco etapas para o desenvolvimento de software: requisitos, arquitetura, implementação, teste e manutenção (SOMMERVILLE, 2011; PRESSMAN, 2005).

Dentre as etapas supracitadas sobre desenvolvimento de software, esta pesquisa irá abordar a Engenharia de Requisitos, que envolve a compreensão e definição dos serviços requisitados do sistema (SOMMERVILLE, 2011). Esta fase é de suma importância nos processos de desenvolvimento uma vez que seu objetivo é realizar o levantamento das necessidades dos *stakeholders* bem como especificar o sistema de forma que os requisitos sejam satisfeitos.

A modelagem orientada por objetivos é uma das pesquisas mais importantes em desenvolvimento na área de Engenharia de Requisitos (FRANCH *et al.*, 2011). Enquanto a modelagem orientada a objeto se encaixa bem nas etapas de análise e projeto de sistemas, a modelagem orientada por objetivos é mais apropriada para os estágios iniciais, onde as metas organizacionais são analisadas para identificar e justificar os requisitos de software e posicioná-los dentro do sistema organizacional (MYLOPOULOS; YU, 1999).

Dentre as linguagens de modelagem baseadas em objetivos propostas, a *Knowledge Acquisition Automated Specification* (KAOS) é uma linguagem usada para modelagem conceitual de objetivos, premissas, agentes, objetos e operações (DARDENNE *et al.*, 1993).

Estender uma linguagem de modelagem é aumentar sua capacidade de representação (BRAMBILLA *et al.*, 2012). As linguagens de modelagem são comumente estendidas como pode ser identificado nos trabalhos de (GONCALVES *et al.*, 2018a), o qual identifica e cataloga as extensões de iStar, e no trabalho de (SOUZA, 2019), o qual identificou as extensões de KAOS por meio de uma *Revisão Sistemática da Literatura* (RSL).

Uma extensão existente geralmente é utilizada para i) modelar aplicações do domínio para o qual foi proposta e ii) ser utilizada como base de uma nova extensão. Uma importante etapa durante este processo é a identificação de uma extensão existente que atenda às necessidades.



Identificar uma extensão de KAOS ou seus construtores com base em suas características, como a área de aplicação, por exemplo, pode ser uma tarefa complexa que requer grande quantidade de tempo e tem uma propensão a não ser exitosa quando executada sem o apoio adequado.

Dois caminhos possíveis para executar este tipo de tarefa sem um catálogo seriam: i) criar uma string de busca, fazer uma busca em bases de dados, selecionar extensões e analisar os resultados ou ii) analisar os artigos disponibilizados pela RSL e tentar identificar a informação que se deseja.

Por outro lado, catálogos têm sido frequentemente utilizados com sucesso para agrupar um conjunto de conhecimentos acumulados, inclusive de extensões de linguagens de modelagem (GONÇALVES *et al.*, 2018b), e permitir buscas.

Um catálogo na engenharia de software pode ser entendido como um compilado de conhecimentos sobre determinado tema que os engenheiros acumulam a partir de experiências anteriores (CHUNG *et al.*, 2012).

Considerando o cenário apresentado, este trabalho tem como problema de pesquisa a necessidade de se catalogar as extensões da linguagem KAOS, de modo a contribuir com a identificação destas extensões e seus construtores.

A solução proposta para este problema pode ser útil aos analistas de requisitos que queiram utilizar uma extensão existente de KAOS ou criar uma nova extensão com base em alguma já existente.

## **1.1 Objetivos**

Nesta seção são apresentados os objetivos gerais e específicos.

### **1.1.1 Objetivo Geral**

- Criar um catálogo online de extensões da linguagem de modelagem KAOS.

### **1.1.2 Objetivos específicos**

- Definir campos e extrair dados das extensões de KAOS e seus construtores;
- Desenvolver e disponibilizar a plataforma do catálogo;
- Avaliar o uso do catálogo para identificar extensões de KAOS e seus construtores.

O resultado deste trabalho poderá ser utilizado para inclusão de novas extensões e para identificar extensões existentes.

## **1.2 Organização do trabalho**

Este trabalho possui a seguinte estrutura: no capítulo 2, os principais conceitos necessários ao entendimento deste trabalho são discutidos; no Capítulo 3, são apresentados trabalhos relacionados, que tratam de estudos na área; o Capítulo 4 descreve a metodologia utilizada neste trabalho; o Capítulo 5 descreve os passos para a criação do catálogo, quais as tecnologia usadas, além de apresentar o catálogo implementado como um dos resultados deste trabalho; no Capítulo 6 é descrito como foi estruturado o estudo exploratório baseados em *survey* que foi feito a fim de avaliar o efeito do uso do catálogo, além dos resultados dessa avaliação; finalmente no Capítulo 7 temos a conclusão deste trabalho e trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

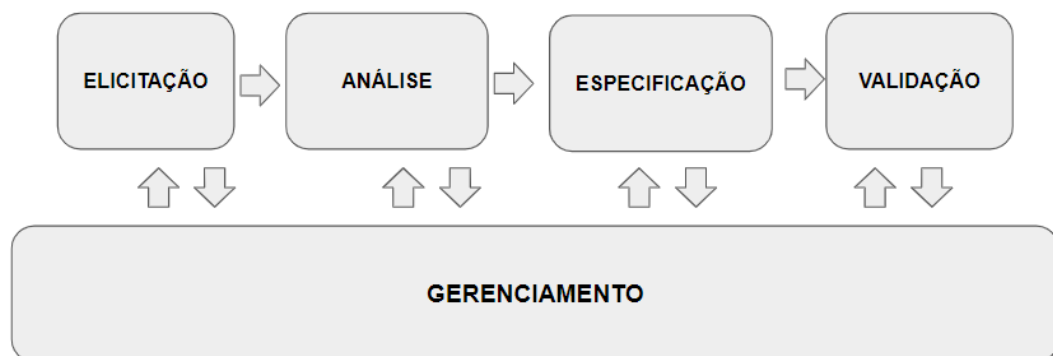
As seções deste Capítulo apresentam os conceitos essenciais para o entendimento deste trabalho.

### 2.1 Engenharia de Requisitos

A Engenharia de Requisitos se concentra na compreensão e definição dos serviços requisitados do sistema. Esta fase é de suma importância nos processos de desenvolvimento uma vez que seu objetivo é realizar o levantamento das necessidades dos *stakeholders* bem como especificar o sistema de forma que os requisitos sejam satisfeito (SOMMERVILLE, 2011).

A Engenharia de Requisitos pode ser entendida como um amplo espectro de tarefas e técnicas que levam ao entendimento dos requisitos. Trata-se de uma ação da Engenharia de Software que se inicia na fase de comunicação e vai até a modelagem, devendo ser adaptada de acordo com a necessidade (PRESSMAN; MAXIM, 2016). Kotonya e Sommerville (1998) dividem as etapas da Engenharia de Requisitos em cinco com o intuito de garantir que os requisitos reflitam as reais necessidades do solicitante, como ilustra a Figura 1.

Figura 1 – Etapas da Engenharia de Requisitos



Fonte: Adaptado de Kotonya e Sommerville (1998)

Na etapa de elicitação são identificadas e coletadas as informações e o escopo do projeto que serão refinados e classificados na etapa de análise. Na especificação, é feita uma descrição das funcionalidades e restrições do sistema a partir do resultado da etapa de análise. Na etapa de validação, os requisitos serão verificados se estão de acordo com as necessidades dos *stakeholders*. Em paralelo a todas as etapas já citadas o gerenciamento cuida da evolução

dos requisitos e rastreamento de novos que possam surgir.

## 2.2 Fundamentos sobre Linguagens de modelagem e extensões

Engenharia Baseada em Modelos (em inglês, *Model-Based Engineering* (MBE)) é um processo no qual os modelos de software desempenham um importante papel, mas não são necessariamente artefatos chave do desenvolvimento. Um exemplo do uso do MBE é o desenvolvimento de um software onde modelos são criados para documentar o sistema. Esses modelos servem como base para a etapa de desenvolvimento, sem envolver nenhuma geração de código automática a partir destes modelos (BRAMBILLA *et al.*, 2012).

Mussbacher *et al.* (2014) definem as linguagens de modelagem como uma maior área de crescimento em MBE. Uma linguagem de modelagem é definida por sua sintaxe abstrata (meta- regras de modelagem e boa formação) e sintaxe concreta. De acordo com Kelly e Tolvanen (2008), as construções de uma linguagem de modelagem devem ser formalizadas e melhor especificadas definindo seu meta-modelo.

Meta-modelo significa modelar sua linguagem mapeando os conceitos da área de aplicação para vários idiomas elementos, suas propriedades e suas conexões, especificadas como links e os papéis que os elementos desempenham neles.

As linguagens de modelagem podem ser classificadas como Linguagens de Modelagem Específicas de Domínio (do inglês, *Domain-Specific Modeling Languages* (DSML)), que possuem construções (Nós e links) para modelar um domínio, e Linguagens de Modelagem de Propósito Geral (do inglês, *General-Purpose Modeling Languages* (GPML)), propostas para modelar qualquer domínio (BRAMBILLA *et al.*, 2012).

A inclusão de novos construtores em uma linguagem de modelagem já existente é chamada de extensão. Uma extensão que envolve a sintaxe abstrata implica introduzir novas metaclasses, propriedades ou relacionamentos no metamodelo ou criar novas regras de boa formação. Enquanto sua representação na sintaxe concreta implica a criação de novas representações gráficas para os novos nós e links, pode também envolver uma representação complementar, como compartimentos ou marcadores textuais (GONCALVES *et al.*, 2018a).

Uma extensão pode ser classificada de acordo com seu impacto na sintaxe original da linguagem, como "conservativa", que mantém a sintaxe original sem alterações, ou "não-conservativa", que altera ou reduz a sintaxe original. Além disso, uma extensão pode incluir uma proposta para representação de um domínio / área de aplicação em particular ou para melhorar

aspectos práticos de uma linguagem (MUSSBACHER *et al.*, 2014).

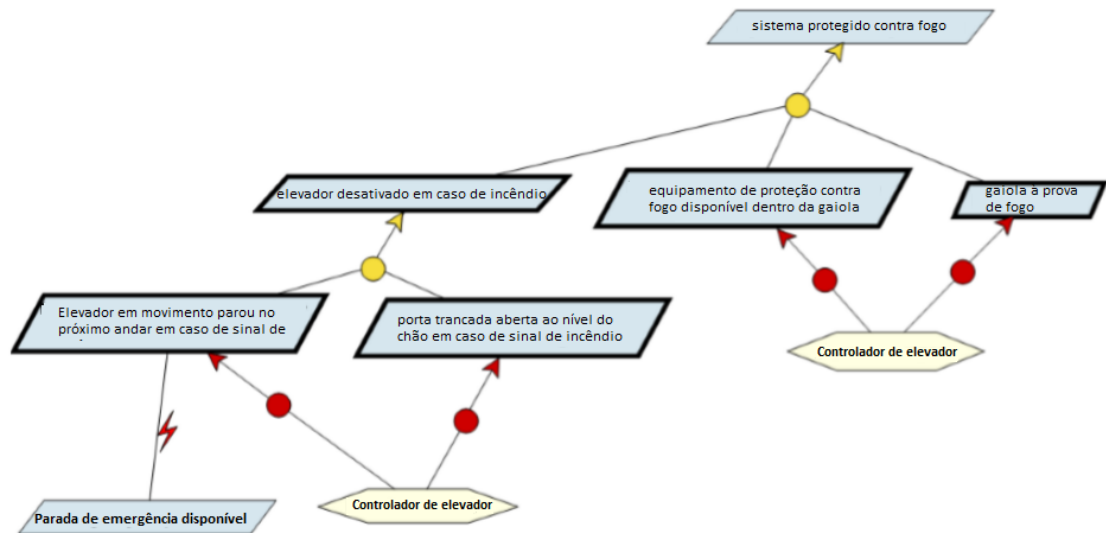
### 2.3 KAOS

KAOS é uma abordagem da Engenharia de Requisitos cujo foco principal é representar os objetivos do sistema, bem como seus conflitos, obstáculos, objetos e quais as consequências do resultado da interação entre os mesmos (HEAVEN; FINKELSTEIN, 2004). KAOS utiliza a combinação de quatro modelos: modelo de objetivo, modelo de responsabilidade, modelo de objeto e modelo de operação. Estes modelos serão detalhados a seguir baseado em (FATIMA, 2015).

- **Modelo de Objetivo:** Este modelo representa os requisitos do sistema como metas e objetivos e, assim, concentrar-se na realização desses objetivos. Objetivos são tipicamente todos os requisitos funcionais e não funcionais que devem ser incorporados no sistema que está sendo desenvolvido, muitas vezes através da assistência de alguns agentes.
- **Modelo de Responsabilidade:** Este modelo envolve entidades chamadas agentes que representam seres humanos ou componentes automatizados e que estão preocupados em atingir os objetivos. O diagrama de responsabilidade descreve para cada agente, os requisitos e expectativas pelas quais ele é responsável ou que foram atribuídos a ele.
- **Modelo de Objeto:** Este modelo se concentra na modelagem dos objetos, entidades, agentes e associações entre eles. As entidades descrevem e traduzem o estado do objeto, mas não realizam operações. Os agentes são responsáveis pela execução das operações. Associações são entidades que dependem do objeto e não têm a capacidade de realizar as operações.
- **Modelo de Operação:** O modelo de operação representa todos os comportamentos que os agentes devem ter para atender suas necessidades. Comportamentos são basicamente operações realizadas por agentes. Na KAOS, um diagrama de operação compreende operações realizadas por um ou vários agentes para atender a um requisito. As composições são feitas por meio de fluxos de dados (a saída de uma operação se torna a entrada de outra operação) ou fluxo de controle (um evento enviado por uma operação aciona ou interrompe outra operação). Um diagrama de operações, portanto, descreve como os agentes precisam cooperar para que o sistema funcione.

A Figura 2 apresenta um exemplo de um modelo de responsabilidade de KAOS para um sistema de proteção contra incêndio de um elevador.

Figura 2 – Exemplo do modelo de responsabilidade KAOS.



Fonte: Respect (2007)

Os hexágonos representam agentes, os paralelogramos são objetivos atribuídos aos agentes e os círculos vermelhos representam responsabilidades. Assim, o agente *Elevator controller* é responsável por manter as portas trancadas quando um alarme de incêndio é acionado e por interromper todos os elevadores em movimento assim que possível. O agente *Elevator company* que instalará o sistema do elevador é responsável por fornecer equipamentos à prova de fogo (gaiola e equipamento).

## 2.4 Extensões de KAOS

Em Semmak *et al.* (2008) os autores propõem uma extensão ao metamodelo de KAOS introduzindo conceitos de variabilidade a fim de poder especificar o Modelo da Família de Requisitos. Esse modelo será habilitado para derivar diferentes modelos específicos de acordo com as necessidades das partes interessadas. Neste trabalho, a proposta de extensibilidade é planejada para o domínio *Cycab*. *Cycabs* são pequenos veículos elétricos, projetados para zonas de acesso restrito (centros históricos da cidade, aeroportos ou estações de trem). As extensões feitas à KAOS devem mostrar o que varia nos sub-modelos KAOS e como isso varia. O que varia é representado graças ao conceito de ponto de variação, e como isso varia é descrito pelos conceitos de faceta e variante. Esses conceitos formam o que é chamado de modelo de variabilidade e o ponto de variação relaciona esse modelo para os sub-modelos KAOS.

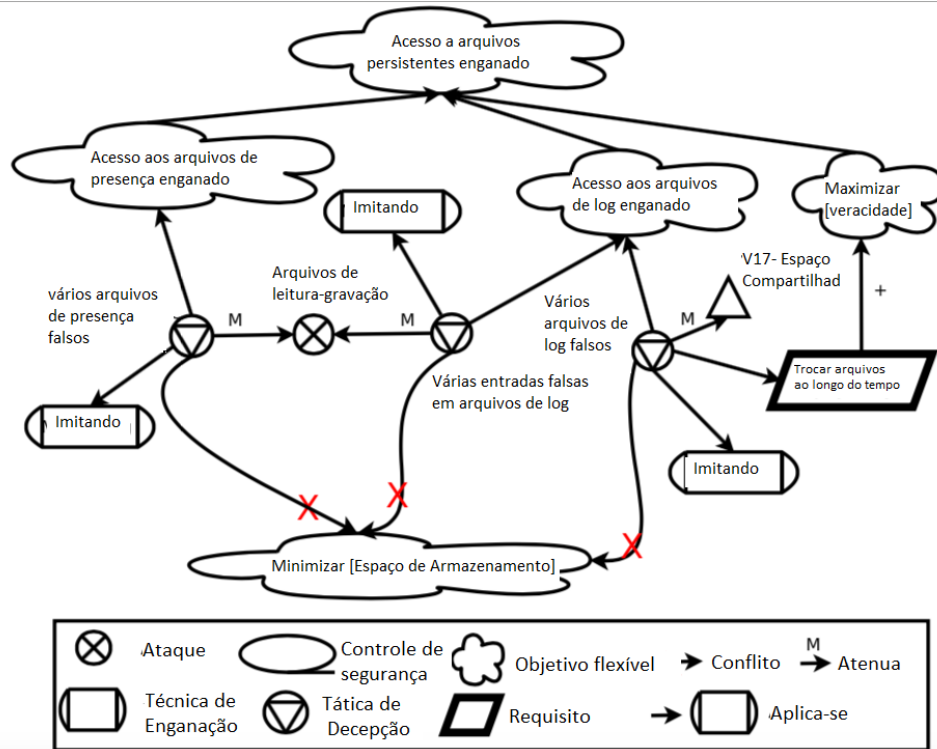
Faveri *et al.* (2016) propõe um método sistemático para incorporar técnicas e táticas de *deception* no design de software. *Deception* é um conceito de segurança de sistemas que

consiste em aplicar técnicas de ilusão de invasores mal intencionados. Ele integra meta e modelos de ameaças construídos durante atividades de ER e produz um modelo de variabilidade expressando um repertório de estratégias de *deception*. Os autores escolheram KAOS para modelar o sistema e-Presence (sistema de controle de presença de alunos para controlar as presenças dos alunos na sala de aula e evitar fraude durante os exames, e que foi requerido pelo conselho de administração da Universidade Nova de Lisboa (FCT/Nova)) porque a linguagem suporta modelagem de ameaças. Na KAOS, os objetivos representam propriedades de um sistema que são satisfeitas por diferentes agentes. O refinamento do objetivo na KAOS cessa quando a responsabilidade da satisfação do objetivo pode ser atribuída a um único agente. Esse objetivo representa um requisito. Objetivos satisfeitos por agentes humanos são tratados como expectativas

A Figura 3 representa três táticas de *deception*. *Multiple Fake Presence Files* e *Multiple Fake Log Files*, as quais objetivam a criação de arquivos de sistema falsos especialmente criados para lançar alertas quando acessados. A técnica de *deception* imita arquivos reais com nomes como “presence.data” e ‘05- 10-2016-12:05.log’. Arquivos de Log precisam ser trocados de periodicamente pois rotatividade de arquivos de log é uma prática comum. Além disso, a técnica *Multiple False Log File Entries* cria registros no arquivo de log com dados sigilosos como usuário / senha. Quando esses dados são usados para acessar algum recurso, um alerta pode ser acionado. Conforme observado no modelo, todas essas táticas de *deception* entram em conflito com o objetivo do sistema Minimizar [espaço de armazenamento].

Lamddi (2017) apresenta uma abordagem para aplicar um método estruturado à integração entre *Safety* e *Security*, criando um modelo de domínio de Segurança e Proteção (SaS). Além disso, demonstra que é possível usar a linguagem orientada a objetivos KAOS na análise de ameaças e perigos para cobrir os domínios de segurança e proteção, fazendo com que saídas, ou artefatos, sejam bem estruturados e abrangentes, o que resulta em confiabilidade devido à abrangência da análise. A abordagem estruturada pode, assim, atuar como uma interface para interações ativas no gerenciamento de risco e perigo em termos de cobertura universal, encontrando soluções para diferenças e contradições que podem ser superadas integrando os domínios de segurança e proteção e usando uma técnica de análise unificada de sistema KAOS que vai resultar na centralidade da análise. Aplicaram o alinhamento entre os modelos de domínio de segurança e proteção (SaS) para o gerenciamento de riscos com a linguagem de modelagem KAOS, o que deu a possibilidade de um melhor método para derivar os requisitos de

Figura 3 – Modelo de táticas de *deception* para autenticação.



Fonte: Favari *et al.* (2016)

segurança e proteção nos estágios iniciais desde o início do ciclo de vida de desenvolvimento do sistema. O alinhamento entre o modelo de domínio SaS e KAOS aumenta a cooperação e facilita a comunicação e a interação entre as partes interessadas.

## 2.5 Catálogos na Engenharia de Software

Uma solução comum na literatura para ajudar os engenheiros de software a cumprirem características de qualidade é o uso de catálogos (CHUNG *et al.*, 2012). De acordo com Cysneiros (2007) citado por Carvalho e Andrade (2019), um catálogo é um conjunto de conhecimentos acumulados com experiência anterior.

Silva *et al.* (2011) define como *Catálogo de Requisitos de Software* (CRS) um conjunto de padrões de requisitos que estão relacionados de forma que venham a sugerir requisitos adicionais. Eles são conhecidos por trazerem melhores especificações dos requisitos permitindo assim a reutilização dos mesmos.

A reutilização de requisitos consiste em utilizar requisitos que foram desenvolvidos para um determinado sistema na modelagem de um sistema diferente. Ao reutilizar requisitos, é possível economizar tempo e esforço, visto que os requisitos reutilizados já foram analisados em outros sistemas (KOTONYA; SOMMERVILLE, 1998). Além disso, o uso de catálogos impede



que os engenheiros gastem tempo pesquisando em fontes diversas ou contando com especialistas da área para tomar decisões sobre como obter requisitos (CHUNG *et al.*, 2012).

Uma nova tendência para o desenvolvimento de software é uso de CRS para uma abordagem de reutilização de requisitos (PACHECO *et al.*, 2015).

Kotonya e Sommerville (1998) afirmam que é possível reusar requisitos em algumas situações, são elas:

1. quando os requisitos estão relacionados ao mesmo domínio de aplicação;
2. quando os requisitos estão relacionados com o mesmo estilo de apresentação de informação;
3. quando os requisitos estão relacionados com políticas organizacionais similares.

### 3 TRABALHOS RELACIONADOS

Para este estudo são considerados trabalhos relacionados, estudos que tem como resultado um agrupamento ordenado de informações relacionadas com modelagem de requisitos no processo de Engenharia de Requisitos.

O trabalho de Goncalves *et al.* (2018a) consiste na identificação e análise das extensões existentes da linguagem de modelagem iStar tendo como base uma RSL que resultou em 96 artigos e 307 construções propostas.

As extensões e construtores foram analisados em três dimensões: análise geral; análise baseada em modelo; e uma terceira dimensão relacionada à clareza semiótica. Nos resultados são apresentadas as extensões do iStar e suas evoluções, segmentadas por áreas de aplicação. Os resultados ainda apontam para a necessidade de um desenvolvimento mais completo, consistente e cuidadoso das extensões do iStar.

Este trabalho se relaciona com o trabalho proposto porque como resultado da RSL é obtido uma lista de trabalhos que definem novas extensões da linguagem de modelagem iStar.

Gonçalves *et al.* (2018b) apresentam um catálogo de extensões de iStar. As extensões e seus construtores foram identificados a partir de uma RSL, que identificou 96 extensões propostas até 2016. O catálogo está disponível em <http://istarextensions.cin.ufpe.br/catalogue/>. Acessando este link é possível ver a lista de todas as extensões do iStar, apresentando as informações de Id, Título, Autores, Ano e Origem.

Este utilizou um gerenciador de conteúdo no formato de aplicação web. Trata-se de uma aplicação implementada usando a linguagem PHP na versão 5.5+ com auxílio do framework Laravel (<https://laravel.com/>). O PHP é uma linguagem de *script open source* de uso geral, muito utilizada, e especialmente adequada para o desenvolvimento web e que pode ser embutida dentro do *HyperText Markup Language* (HTML). Já o Laravel é um *framework* de desenvolvimento rápido para PHP, livre e de código aberto. O Laravel utiliza o Composer para gerenciar suas dependências, algo que praticamente toda aplicação PHP moderna faz.

É possível pesquisar as extensões do iStar por autor e título ou filtrar as extensões por área de aplicação, abordagem de raciocínio usada, tipo de construções propostas, tipo de validação, nível de extensão, suporte à ferramenta, abordagem de raciocínio usada e definição de regras de formação. Também é possível pesquisar construções pelo nome da construção, área de aplicação, forma utilizada, tipo de construção e objetivo da representação.

Este trabalho se relaciona com o trabalho proposto porque tem como resultado uma

ferramenta online (catálogo) com a lista de trabalhos que definem novas extensões da linguagem de modelagem iStar obtidas a partir de uma RSL.

O trabalho de Carvalho *et al.* (2018) visa capturar subcaracterísticas e soluções para Invisibilidade e catalogá-las em um *Softgoal Interdependency Graph* (SIG), que é um formato conhecido para catalogar *Requisitos Não-Funcionais* (RNFs). Como não existe uma abordagem sistemática sobre como construir SIGs, também foi proposto sistematizar a definição de SIG de Invisibilidade usando os seguintes métodos de pesquisa bem definidos: *snowballing*, pesquisa em banco de dados, teoria fundamentada em questionários.

Como resultado, foi obtido um SIG de invisibilidade composto por duas subcaracterísticas principais, doze subcaracterísticas, dez soluções gerais e cinquenta e seis soluções específicas. Esses conhecimentos são úteis para apoiar os engenheiros de software a especificar requisitos e soluções práticas para aplicativos UbiComp e IoT. Além disso, a metodologia proposta usada para capturar e catalogar requisitos em um SIG pode ser reutilizada para outros NFRs.

Este trabalho se relaciona com o trabalho proposto porque tem como resultado um SIG que captura requisitos não-funcionais flexíveis que são refinadas e catalogados em requisitos menores, a partir das quais derivam as possíveis soluções de design conhecidas como operacionalizações.

A RSL de extensões de KAOS (SOUZA, 2019) tem o objetivo de identificar e analisar as extensões de KAOS. A busca da RSL identificou 955 artigos em 7 bases de dados eletrônicas. Deste total, 22 artigos foram selecionados, os quais definem novas extensões de KAOS. As extensões identificadas e seus construtores foram analisados de acordo com questões de pesquisa definidas, apresentando a evolução das extensões ao longo dos anos, principais autores, locais de publicação, dentre outras análises. Uma atualização desta RSL que vem sendo desenvolvida como parte de um trabalho de mestrado do *Programa de Pós-Graduação em Computação* (PCOMP) da UFC - Quixadá (<https://pcomp.quixada.ufc.br>) mostrou um aumento de 20 novas extensões. Deste modo, consideramos o total de 42 extensões.

Este trabalho se relaciona com o trabalho proposto pois os resultados da RSL são utilizados para a extração de dados a serem utilizados para popular o catálogo.

Tabela 1 – Comparação entre os trabalhos relacionados e o proposto.

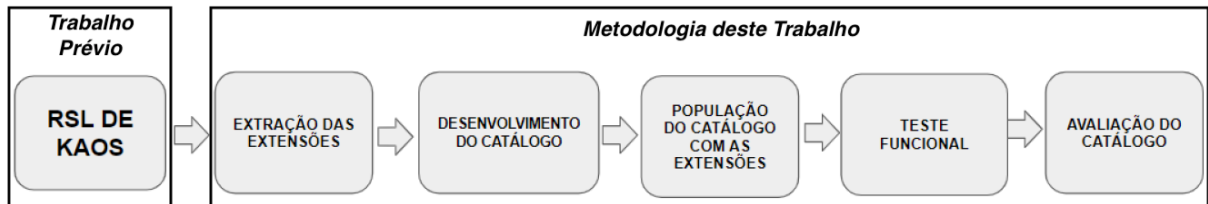
<b>Critério</b>	<b>(GONCALVES <i>et al.</i>, 2018a)</b>	<b>(GONÇALVES <i>et al.</i>, 2018b)</b>	<b>(CARVALHO <i>et al.</i>, 2018)</b>	<b>(SOUZA, 2019)</b>	<b>Trabalho Proposto</b>
É Catalogo		X	X		X
Agrupar Extensões	X	X		X	X
Permite busca	X				X
Kaos				X	X

Fonte: Elaborado pela autor, (2019)

## 4 METODOLOGIA

Nesta seção são descritos os passos realizados para se atingir o objetivo deste trabalho. A Figura 4 apresenta a sequência destes passos.

Figura 4 – Metodologia usada para criação do catálogo de extensões do KAOS



Fonte: Elaborado pelo Autor, (2022)

Os resultados da RSL de extensões de KAOS (SOUZA, 2019) são utilizado como ponto de partida dos passos executados neste trabalho. Assim, este foi representado na Figura 4, não como um passo da metodologia, mas como um resultado anterior que tem uma relação com o método utilizado.

Cada passo da metodologia é descrito nas sessões a seguir.

### 4.1 Extração das Extensões

Para extrair as informações das extensões assim como as informações dos construtores foram construídos templates que serviram como base para a extração dos dados das extensões, criação e população do catálogo. Os templates são apresentados na seção 5.1.

### 4.2 Desenvolvimento do catálogo

O catálogo de extensões de KAOS foi desenvolvido como uma solução web baseada nos campos do formulário de extração descrito no passo anterior. Foi seguida uma abordagem incremental, iniciando-se pelas funcionalidades relacionadas às extensões e em seguida foram implementadas as funcionalidades relacionadas aos construtores. O catálogo está disponível em <https://kaos-catalogue-d9e49.web.app>. Ele é apresentado na 5.2, juntamente com o detalhamento

das decisões tecnológicas envolvidas.

### 4.3 População do Catálogo

A partir da planilhas<sup>1</sup> obtidas no passo de Extração das Informações e com o *deploy* do catálogo feito no *Firebase Hosting*, as extensões, assim como seus construtores foram cadastradas no catálogo online.

### 4.4 Teste Funcional do Catálogo

Teste funcional é um processo de garantia de qualidade e um tipo de teste de caixa preta que baseia seus casos de teste nas especificações do componente de software sob teste. As funções são testadas alimentando-as com entrada e examinando a saída.(COSTEN, 2019)

As funcionalidades do catálogo foram testadas com base em uma planilha de teste. Os testes foram realizados pelo autor deste trabalho, por um aluno de mestrado em ciência da computação e por um professor especialista em requisitos.

Os *bugs* identificados foram corrigidos e as melhorias apontadas foram implementadas a fim de criar uma melhor experiência de usuário no uso do catálogo.

O teste funcional é detalhado na seção 5.5.

### 4.5 Avaliação do Catálogo

Um estudo exploratório foi realizado a fim de avaliar a corretude da identificação de informações de extensões por meio do catálogo de KAOS e o esforço envolvido. Para tanto, foi aplicado um questionário baseado nos princípios de (KITCHENHAM; PFLEEGER, 2002). A avaliação do catálogo é detalhada no capítulo 6.

---

<sup>1</sup> Link para as planilhas: <http://bit.do/fTpM8>

## 5 CATÁLOGO DE EXTENSÕES DA LINGUAGEM DE MODELAGEM KAOS

O catálogo de extensões da KAOS será apresentado nas seções a seguir. Este capítulo está estruturado da seguinte forma: Inicialmente descreveremos a extração das extensões por meio dos templates, as tecnologias envolvidas no desenvolvimento e em seguida uma visão geral.

### 5.1 Extração das Extensões

Foram criados dois templates, um para extração das informações das extensões e outro para extração das informações dos construtores onde foram considerados os campos que foram extraídos na RSL.

A lista dos campos do template das informações das extensões é apresentada na Tabela 2.

Tabela 2 – Informações extraídas das extensões

---

ID - Código para identificar a extensão
Título - Título do artigo relacionado à extensão KAOS
Link - Link para acessar o documento da extensão KAOS
Autores - Autores do artigo relacionados à extensão KAOS
Local onde foi publicado - (Jornal / Conferência / Livro)
Extensão Base - Se a extensão for baseada em outra extensão, essas informações os une.
Área de aplicação - Representa a área de aplicação em que a extensão está proposta. Pode ser um ou mais dos seguintes valores (Social, Inteligente, Segurança, Contextual, Empresa, Desenvolvimento Geral, Linhas de Produtos de Software, Aspectos e outros requisitos não funcionais)
Nível - O nível de representação envolvido na proposta da extensão (sintaxe abstrata, sintaxe concreta ou ambas)
Compatibilidade com o metamodelo - Para identificar se há compatibilidade entre os representantes construção de novas construções no metamodelo e concreto sintaxe

---

Fonte: Elaborado pela autor, (2022)

A lista dos campos do template das informações dos construtores é apresentada na Tabela 3.

Tabela 3 – Informações extraídas dos Construtores

ID - Código para identificar a construção
Extensão relacionada Objetivo - A extensão à qual a construção está relacionada
Descrição
Significado
Tipo - Link ou Nó
Exemplo

Fonte: Elaborado pela autor, (2022)

As informações foram extraídas manualmente dos artigos das extensões. O catálogo foi desenvolvido levando em conta essa estrutura e os dados extraídos foram utilizados para preencher o catálogo.

## 5.2 Tecnologias

O catálogo de extensões do KAOS foi desenvolvido com base nas tecnologias listadas na Tabela 4.

Tabela 4 – Tecnologias utilizadas

Finalidade da Tecnologia	Tecnologia Utilizada
Plataforma de Desenvolvimento	<i>Firestore</i>
Prover autenticação	<i>Firestore Authentication</i>
Ambiente de <i>deploy</i>	<i>Firestore Hosting</i>
Banco de dados	<i>Firestore Realtime Database</i>
Armazenamento de arquivos	<i>Firestore Cloud Storage</i>
Biblioteca para criação de interfaces	<i>React</i>
Linguagem de programação	<i>Typescript</i>

Fonte: Elaborado pela autor, (2022)

A seguir descrevemos cada uma destas tecnologias.

O **Firestore** (<https://firebase.google.com/>) fornece ferramentas para desenvolver aplicativos web de alta qualidade. Ele foi escolhido por ter uma estrutura pronta, segura, de rápida implementação, facilmente escalável e gratuita. Dentre estas ferramentas mencionadas, selecionamos as seguintes:

- **Firestore Authentication:** fornece serviços de back-end, *Software Development kits* (SDKs) fáceis de usar e bibliotecas de *User Interface* (UI) prontas para autenticar usuários no seu aplicativo.



- **Hosting:** é um recurso de hospedagem de conteúdo da Web de nível de produção para desenvolvedores.
- **Realtime Database:** é um banco de dados NoSQL hospedado na nuvem. Os dados são armazenados como *JavaScript Object Notation* (JSON) e sincronizados em tempo real para cada cliente conectado. Os bancos de dados *Not Only Structured Query Language* (NoSQL) (ou não-relacionais) utilizam um padrão diferente de armazenamento em relação ao *Structured Query Language* (SQL) onde não há necessidade de agrupar os dados em um esquema de tabelas para usar as informações. No apêndice B é possível ver como o JSON é estruturado no console do *Realtime Database*.
- **Cloud Storage:** é um serviço de armazenamento de objetos avançado, simples e econômico criado para a escala do *Google*. Com os SDKs do *Firebase* para *Cloud Storage*, é possível usar a segurança do *Google* para fazer *upload* e *download* de arquivos nos apps do *Firebase*, independentemente da qualidade da rede. O *Cloud Storage* foi usado para fazer o *upload* das imagens que representam os exemplos gráficos de cada construtor. No apêndice B é possível ver como o console do *Cloud Storage* é estruturado.

Adicionalmente, utilizamos o **React** (<https://reactjs.org/>), que se trata de uma biblioteca *JavaScript* (<https://www.javascript.com/>) declarativa, eficiente e flexível para construir interfaces de usuário. Ele permite que você componha interfaces de usuário complexas a partir de pedaços pequenos e isolados de código chamados de “componentes”. Para criar uma aplicação *React* basta ter uma versão recente do Node.JS instalado e usar o comando “`npx-create-react-app nome-do-app`” no terminal. O Node.js pode ser definido como um ambiente de execução *Javascript* server-side.

A linguagem de programação escolhida foi o **TypeScript** que é uma linguagem de programação fortemente tipada que se baseia em *JavaScript*, oferecendo melhores ferramentas em qualquer escala. O *TypeScript* adiciona sintaxe adicional ao *JavaScript* para oferecer suporte a uma integração mais estreita com o editor. O código *TypeScript* é convertido em *JavaScript*, que é executado em qualquer lugar que o *JavaScript* seja executado. O *TypeScript* entende *JavaScript* e usa a inferência de tipo para fornecer ótimas ferramentas sem código adicional.

O uso do *TypeScript* traz a robustez de uma tipagem estática, o que torna o projeto muito mais escalável, conciso e com a possibilidade de identificar bugs ainda em tempo de

compilação.

Por fim, o código do catálogo está disponível no *github* no link <https://github.com/pedrojunioor/KaosCatalogue> e é livre para reutilização ou adaptação para propósitos similares.

### 5.3 O Catálogo

O catálogo de extensões de KAOS foi proposto para facilitar a identificação de extensões de KAOS e seus construtores com base em suas características, como a área de aplicação ou tipo de construtor. Nesta seção apresentamos uma visão geral da ferramenta.

#### 5.3.1 Visão Geral

O catálogo guarda as seguintes informações sobre extensões: o título, área de aplicação, autores, ano de publicação, nome do periódico / congresso, se é extensão derivativa e se sim de qual ela deriva compatibilidade entre metamodelo e sintaxe concreta de extensões, tipo de validação, se apresenta a definição dos conceitos, se há suporte de ferramenta, nível de extensão, completude do metamodelo, lista de construtores além de um link que leva para site de onde o trabalho foi publicado.

São apresentadas as seguintes informações sobre os construtores: descrição, significado, formato e exemplo visual.

A ferramenta permite a busca de extensões por autor ou título e permite filtrar por área de aplicação, ano de publicação, local de publicação ou se ela tem ferramenta de suporte.

É possível ver a lista de todas as extensões catalogadas e selecionar alguma para detalhar mostrando todas as informações supracitadas além de novas informações incluindo um link para acessar o local de publicação. A lista dos construtores que pertencem aquele estudo também é exibida com um botão para acessar os detalhes daquele construtor.

Similar à lista de extensões também é possível acessar a lista de todos os construtores catalogados e selecionar algum para ver suas informações detalhadas. É possível buscar um construtor por nome ou se trata-se de um construtor Link ou Nó. Na página de detalhes da extensão é possível ver a qual extensão aquele construtor pertence e ao clicar no título da extensão a aplicação é redirecionada para a página de detalhes dessa extensão.

Por fim, é possível reportar uma nova extensão informando o título, o link para acessá-

la e o e-mail. As extensões informadas podem ser endossadas pela aceitação de especialistas em extensões KAOS.

## 5.4 Apresentação do Catálogo

A Figura 5 mostra a página inicial do catálogo com um texto de apresentação. Nesta tela assim como em todas as outras existem os links para ir até a página de listagem das extensões, listagem dos construtores ou a página para reportar uma nova extensão. Para reportar uma nova extensão é necessário fazer login na plataforma com uma conta *Google*.

Figura 5 – Página Inicial



Fonte: Elaborado pelo Autor, (2022)

A Figura 6 mostra a lista de extensões cadastradas com informações prévias e um link para acessar as informações detalhadas de cada extensão. É nesta página que é possível buscar as extensões por autor ou título ou ainda filtrar por área de aplicação, ano de publicação, local de publicação ou se ela tem ferramenta de suporte.

Figura 6 – Página de Extensões

KAOS Catalogue G Login

Extensions Constructs Form Report

Title  Search

Extensions					
-	Title	Author	Date	Source	
1	A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty	Betty H.C. Cheng, Pete Sawyer, Nelly Bencomo e Jon Whittle	2009	conference	<a href="#">Detail</a>
2	Analysis and Verification of Time Requirements Applied to the Web Services Composition	Gregorio Díaz, María-Emilia Cambroner, M. Llanos Tobarra, Valentin Valero e Fernando Cuartero	2006	workshop	<a href="#">Detail</a>
3	AspectKAOS- Integrating Early-Aspects into KAOS	André Gil e João Araújo	2009	workshop	<a href="#">Detail</a>
4	Assessing requirements-related risks through probabilistic goals and obstacles	Antoine Cailliau e Axel van Lamsweerde	2013	journal	<a href="#">Detail</a>

Fonte: Elaborado pelo Autor, (2022)

A Figura 7 mostra a página de detalhamento de uma extensão. Nela é possível ver todas as informações sobre aquela extensão, um link para o local onde a extensão foi publicada, além da lista de construtores que venham a pertencer a esta extensão. Na lista de construtores também é possível acessar os detalhes de cada um e ver um exemplo gráfico na maioria dos construtores catalogados.

Figura 7 – Página de detalhamento de uma extensão

KAOS Catalogue Junior Carvalho G Logout

Extensions Constructs Form Report

A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty

Application Area Adaptive Systems	Author Betty H.C. Cheng, Pete Sawyer, Nelly Bencomo e Jon Whittle	Year Publication 2009	Extension Derivative not
Extension Base	Source conference	Source Location Model Driven Engineering Languages and Systems	Validation Form case-study
Metamodel Completeness complete	Syntax level concrete-and-abstract	Tool Support not	Definition of Concepts definition is presented

Fonte: Elaborado pelo Autor, (2022)

A Figura 8 mostra o formulário para o report de uma nova extensão. Depois de feito o login a extensão pode ser reportada e o administrador da plataforma tem a opção de aceitá-la ou não.

Figura 8 – Página de Report

Fonte: Elaborado pelo Autor, (2022)

A Figura 9 mostra uma página que é visível apenas ao usuário com permissão de administrador. Nela são listadas todas as extensões que foram reportadas e o link para acessá-las. Depois de feita a análise e decidir se trata-se ou não de uma nova extensão do KAOS o administrador pode rejeitar e aquele report e este é excluído da lista, ou pode aceitá-lo e ao clicar em aprovar, o formulário de cadastro para a nova extensão é aberto.

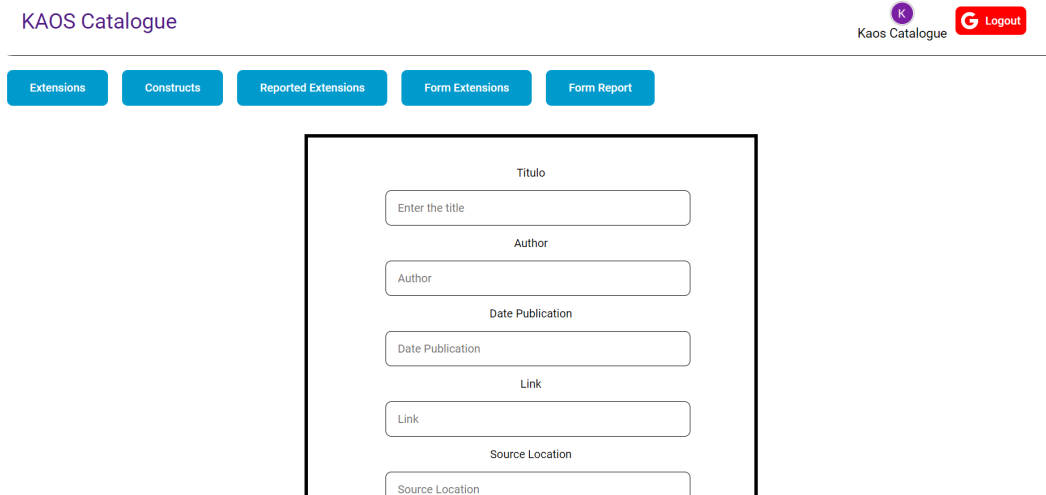
Figura 9 – Extensões Reportadas

Reported Extensions						
-	Title	Author	Link	Reported By	Status	To approve
1	Titulo Teste	Author Teste	teste	kaoscatalogue@gmail.com	-	Approve Reject

Fonte: Elaborado pelo Autor, (2022)

A Figura 10 mostra a página do formulário para cadastro de uma nova extensão de KAOS. Este formulário é visível apenas para o usuário administrador.

Figura 10 – Formulário de Cadastro de Extensões



Fonte: Elaborado pelo Autor, (2022)

### 5.5 Teste Funcional do Catálogo

Nesta seção apresentamos detalhes dos casos de testes funcionais relacionados ao catálogo. Inicialmente foi criada uma planilha de teste, a qual está disponível em <https://bityli.com/MFXgq>. Os testes foram realizados pelo autor deste trabalho, por um aluno do mestrado em computação (PCOMP - pcomp.quixada.ufc.br) e por um professor especialista em requisitos.

O status de cada funcionalidade testada foi reportado bem como necessidade de mudanças e sugestões de melhorias.

Cada funcionalidade a ser testada tem um objetivo definido, pré-requisitos para que a funcionalidade possa ser cumprida, cenários e status para definir se a funcionalidade está se comportando conforme o esperado.

A Figura 11 mostra um exemplo retirado da planilha de testes de tarefa a ser executada a fim de testar o fluxo de reportar uma nova extensão.

Figura 11 – Exemplo de fluxo a ser testado

<b>Funcionalidade 5 - Reportar uma nova extensão</b>	
Objetivo da Funcionalidade 5: Reportar uma nova extensão que ainda não está cadastrada no catálogo	
Funcionamento da Funcionalidade 5: Acessar a página inicial do catálogo, clicar em no botão com label Form Report. O catálogo é redirecionado a uma nova página. No centro da página é exibido um formulário com 3 inputs. No primeiro input informe o título da nova extensão, no segundo input informe o autor, no terceiro input informe o link onde a o estudo que definiu uma nova extensão foi publicado. Finalizar a ação clicando no botão Report.	
Pré-requisito para que o objetivo seja alcançado: Acesso a internet em um ambiente desktop, ter uma conta google e fazer login na plataforma usando essa conta.	
1) Possíveis resultados do Teste:	
<Cenário da funcionalidade>	<Descrição do cenário>
Dados válidos	Ao preencher os 3 campos e clicar em Report, um caixa de diálogo deve aparecer confirmando o report da extensão
Dados inválidos	Se algum dos campos não forem preenchidos, a ferramenta não permite o Report.
STATUS	OK

Fonte: Elaborado pelo Autor, (2022)

A Figura 12 mostra a página de report de uma nova extensão com os dados preen-

chidos e a Figura 13 mostra a lista de novas extensões reportadas no cenário que o report foi confirmado pelo usuário.

Figura 12 – Reportar uma nova extensão

Fonte: Elaborado pelo Autor, (2022)

Figura 13 – Extensões Reportadas

Reported Extensions						
-	Title	Author	Link	Reported By	Status	To approve
1	TESTE DE REPORT	JUNIOR CARVALHO	www.google.com	kaoscatalogue@gmail.com	-	Approve Reject

Fonte: Elaborado pelo Autor, (2022)

A Figura 14 descreve uma tarefa de verificação de consistência de dados. No cenário em que dados válidos são fornecidos para a busca, o catálogo deve exibir os dados de acordo com a planilha de extrações.

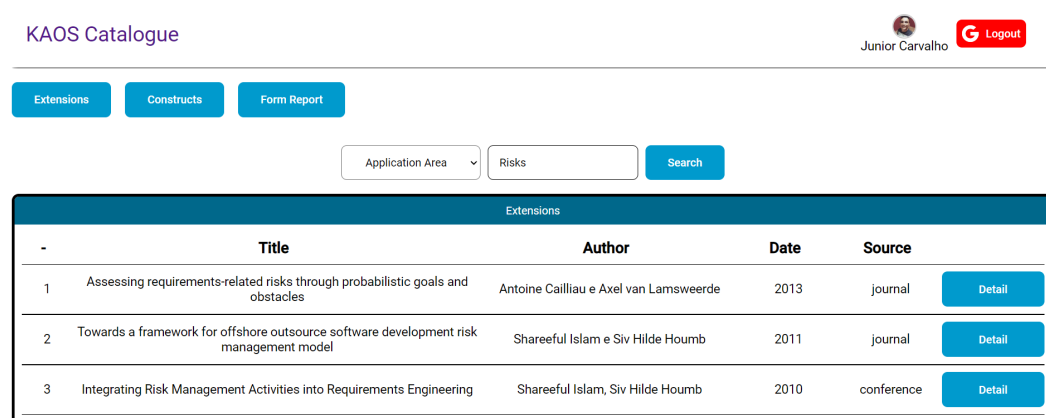
Figura 14 – Exemplo de funcionalidade a ser testada

Funcionalidade 3 - Filtrar extensões por área de aplicação	
Objetivo da Funcionalidade 3: Verificar os títulos pertencentes a uma área de aplicação específica	
Funcionamento da Funcionalidade 1: Acessar a página inicial do catálogo, clicar em no botão com label Extensions. O catálogo é redirecionado a uma nova página. No topo desta página dois inputs são exibidos além da lista de todas as extensões cadastradas no catálogo. No primeiro input selecionar a opção "Application Area" no input de filtros, em seguida digitar a área de aplicação pela qual as extensões devem ser filtradas. Finalizar a ação clicando no botão SEARCH.	
Pré-requisito para que o objetivo seja alcançado: Acesso a internet em um ambiente desktop	
1) Possíveis resultados do Teste:	
< Cenário da funcionalidade >	< Descrição do cenário >
Dados válidos	Selecionar a opção "Application Area" no primeiro input. Inserir "Risks" e clicar em SEARCH o catálogo deve exibir 3 extensões com link para acessar os detalhes de cada uma.
Dados inválidos	Selecionar a opção "Application Area" no primeiro input. Inserir "Testes" e clicar em SEARCH o catálogo não deve exibir nenhuma extensão.
STATUS	OK

Fonte: Elaborado pelo Autor, (2022)

De acordo com a planilha de extrações existem 3 (três) extensões propostas pertencentes a área de aplicação Risks. A Figura 15 mostra que ao aplicar o filtro Application Area e inserir a string Risks o catálogo exibe 3 (três) extensões.

Figura 15 – Resultado ao filtrar as extensões pela área de aplicação Riscos



The screenshot shows the KAOS Catalogue interface. At the top, there are navigation buttons for 'Extensions', 'Constructs', and 'Form Report'. Below these, there is a search bar with a dropdown menu for 'Application Area' and a text input field containing 'Risks'. A 'Search' button is next to the input field. The search results are displayed in a table with the following columns: Title, Author, Date, and Source. Each row also has a 'Detail' button.

Extensions				
-	Title	Author	Date	Source
1	Assessing requirements-related risks through probabilistic goals and obstacles	Antoine Cailliau e Axel van Lamsweerde	2013	journal
2	Towards a framework for offshore outsource software development risk management model	Shareeful Islam e Siv Hilde Houmb	2011	journal
3	Integrating Risk Management Activities into Requirements Engineering	Shareeful Islam, Siv Hilde Houmb	2010	conference

Fonte: Elaborado pelo Autor, (2021)

No decorrer dos testes a necessidade de 9 (nove) correções foram descobertos e corrigidas. Todas elas eram relacionadas a melhorar ou permitir um fluxo esperado no catálogo.

## 5.6 Limitações

Nesta seção listamos algumas limitações do catálogo dentre elas algumas percebidas pelos participantes do estudo. A primeira seria a falta de um *input* genérico para filtrar as extensões, onde ao inserir qualquer string nesse campo ela seria comparada com todas as informações das extensões sem a necessidade de escolher um tipo de filtro específico. Outra limitação é que o catálogo não foi implementado 100% responsivo então ao acessá-lo em um ambiente mobile a ferramenta pode não se comportar da forma devida. Não manter os



filtros aplicados ao navegar dentro da plataforma também foi outra limitação percebida pelos participantes do estudo.

## 6 AVALIAÇÃO

Este capítulo descreve a avaliação do catálogo. Na Seção 6.1 é descrita a metodologia desta avaliação, na Seção 6.2 são apresentados os perfis dos participantes, na Seção 6.3 são descritos os resultados obtidos e discussões e, por fim, na Seção 6.5 são apresentadas as ameaças à validade.

### 6.1 Metodologia da Avaliação

[Objetivo] O propósito deste estudo é analisar a corretude da identificação de informações de extensões por meio do catálogo de KAOS e o esforço envolvido.

[Caracterização do estudo] Foi realizado um estudo exploratório com coleta de dados baseada em questionário (survey) quanti-qualitativo seguindo os princípios de (KITCHENHAM; PFLEEGER, 2002).

[População e amostra] Acreditamos que pesquisadores experientes podem se beneficiar do uso do catálogo. Entretanto, acreditamos que os usuários sem experiência (naive) estão mais propensos a erros e ter uma maior dificuldade para encontrar extensões e seus construtores. Por conta disto, definimos participantes sem experiência com extensões de KAOS como população alvo do estudo. Escolhemos os alunos de uma turma do curso de Engenharia de Software da Universidade Federal do Ceará em Quixadá como amostra. Vinte e dois (22) alunos foram convidados, sendo que dezessete (17) participaram.

[Preparação da Coleta] Foi criado um questionário por meio do *Google Forms*. A lista de questões está disponível no apêndice A.

Na primeira pagina do questionário é exibido um texto de esclarecimento e livre consentimento.

O questionário usado nesta avaliação está organizado em quatro partes:

1. Caracterização do perfil do participante;
2. Perguntas relacionadas às extensões do catálogo;
3. Perguntas relacionadas aos construtores do catálogo;
4. Perguntas relacionadas ao uso do catálogo de forma geral.

Cada item das partes 2 e 3 envolve a execução de uma tarefa com o catálogo e possui um par de perguntas quantitativas associado. Estas perguntas analisam o resultado da tarefa solicitada e o nível de dificuldade para realizá-la. A Figura 16 mostra um exemplo.

Figura 16 – Exemplo de tarefa do questionário

2.1. Consultar Extensão

O trabalho "A comparison of goal-oriented approaches to model software product lines variability" é uma extensão de KAOS? \*

Sim

Não

Qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada acima? \*

MUITO FÁCIL    1    2    3    4    5    MUITO DIFÍCIL

Fonte: Elaborado pelo Autor, (2022)

Além de perguntas qualitativas que dizem respeito a percepção da dificuldade encontrada para responder cada questão.

Antes de iniciar avaliação, foi executado um teste piloto com 4 (quatro) participantes. O teste piloto contribui para aprimorar os planos para a coleta de dados tanto em relação ao conteúdo dos dados quanto aos procedimentos que devem ser seguidos (YIN, 2015). Ele ajuda a validar o instrumento de pesquisa projetado a medida que é aplicado antes da entrar em contato com os participantes designados para o estudo. As correções identificadas no piloto foram realizadas e os dados das respostas foram descartados.

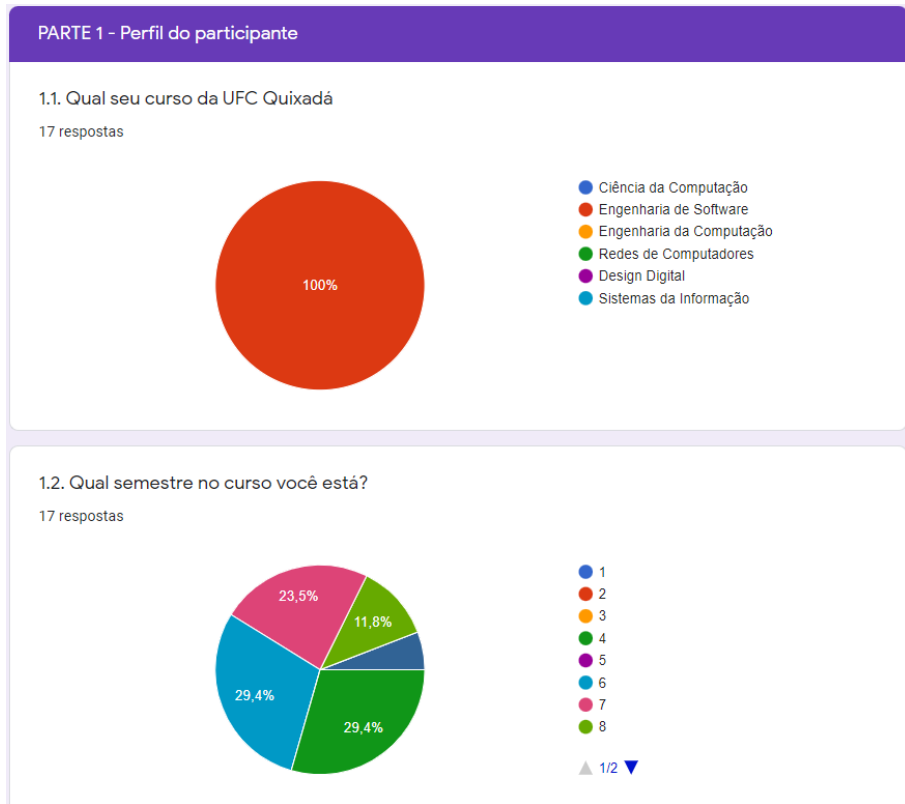
[Execução do Estudo] Devido a pandemia de COVID-19, a realização deste estudo foi realizado em 27 de janeiro de 2022 de forma online por meio do *Google Meet* e seguiu a seguinte dinâmica: Inicialmente o catálogo foi apresentado por meio de slides e uma demonstração de uso. Dúvidas dos participantes foram sanadas e em seguida foi solicitado que os mesmos respondessem ao questionário.

## 6.2 Caracterização do Perfil dos Participantes

A primeira parte do questionário diz respeito a caracterização do perfil dos participantes. A Figura 17 apresenta os resultados das primeiras questões sobre o perfil. Ela mostra que todos os participantes são alunos do curso de Engenharia de Software, mas estão em pelo

menos 5 semestres distintos.

Figura 17 – Curso e semestre

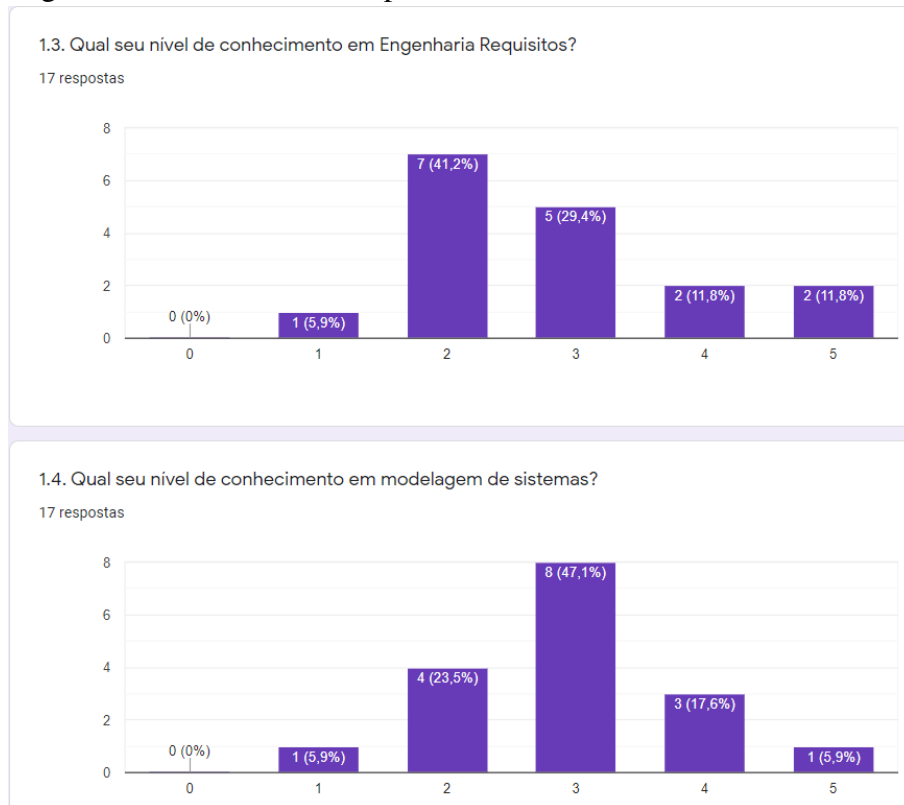


Fonte: Elaborado pelo Autor, (2022)

No formulário cada participante deveria informar também qual seu nível de conhecimento prévio sobre Engenharia de Requisitos e qual seu nível de conhecimento prévio sobre Modelagem de Sistemas. A resposta partia de 0 (sem conhecimento) até 5 (especialista).

Na Figura 18 ainda são exibidos os resultados de questões sobre o perfil dos participantes. Nela podemos ver que a maioria dos alunos já possuíam algum conhecimento prévio tanto sobre Engenharia de Requisitos quanto sobre Modelagem de Sistemas.

Figura 18 – Conhecimentos prévios



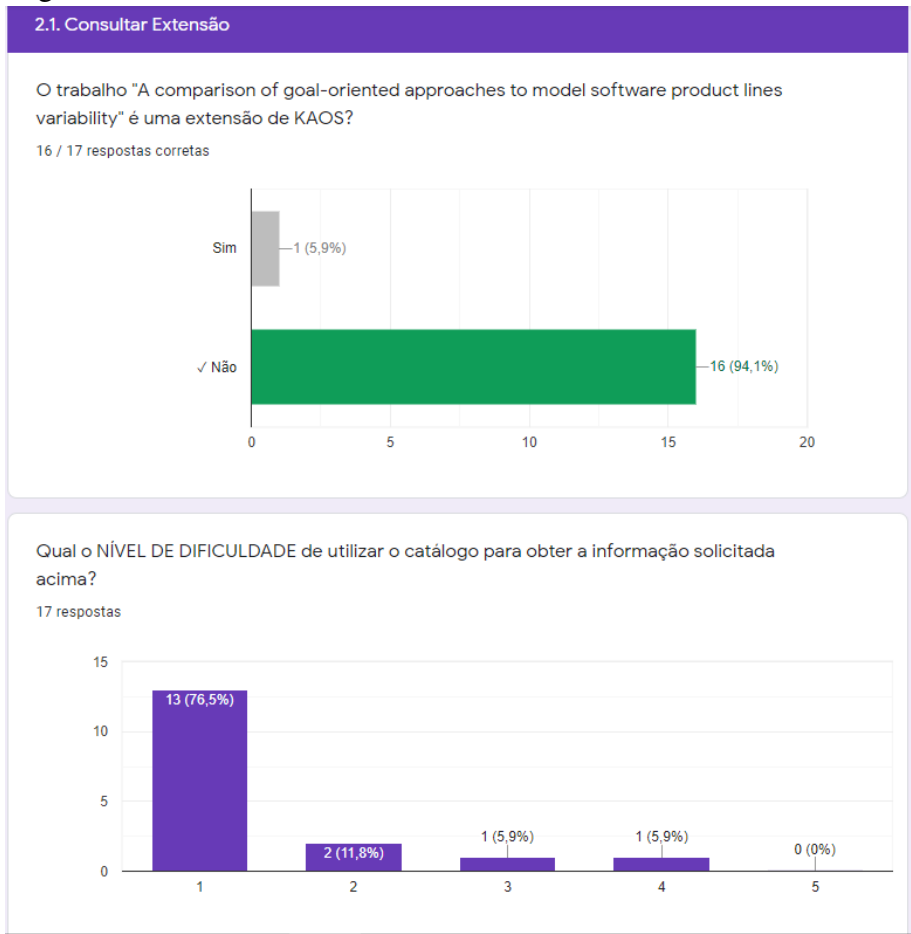
Fonte: Elaborado pelo Autor, (2022)

### 6.3 Resultados da Avaliação

A segunda parte do questionário traz as questões que envolvem e avaliam o uso do catálogo de extensões de KAOS. Apresentaremos os resultados destas questões nesta seção.

A Figura 19 mostra os resultados obtidos na primeira pergunta do questionário. Os participantes usaram o catálogo para responder a questão proposta e em seguida informaram o grau de dificuldade para responde-la como no exemplo da Figura 16. Podemos ver um percentual de aproximadamente 100% de acertos. Aliado a esse alto percentual de acertos, também é possível ver que a maioria dos participantes julgou como muito fácil ou fácil obter a resposta usando o catálogo.

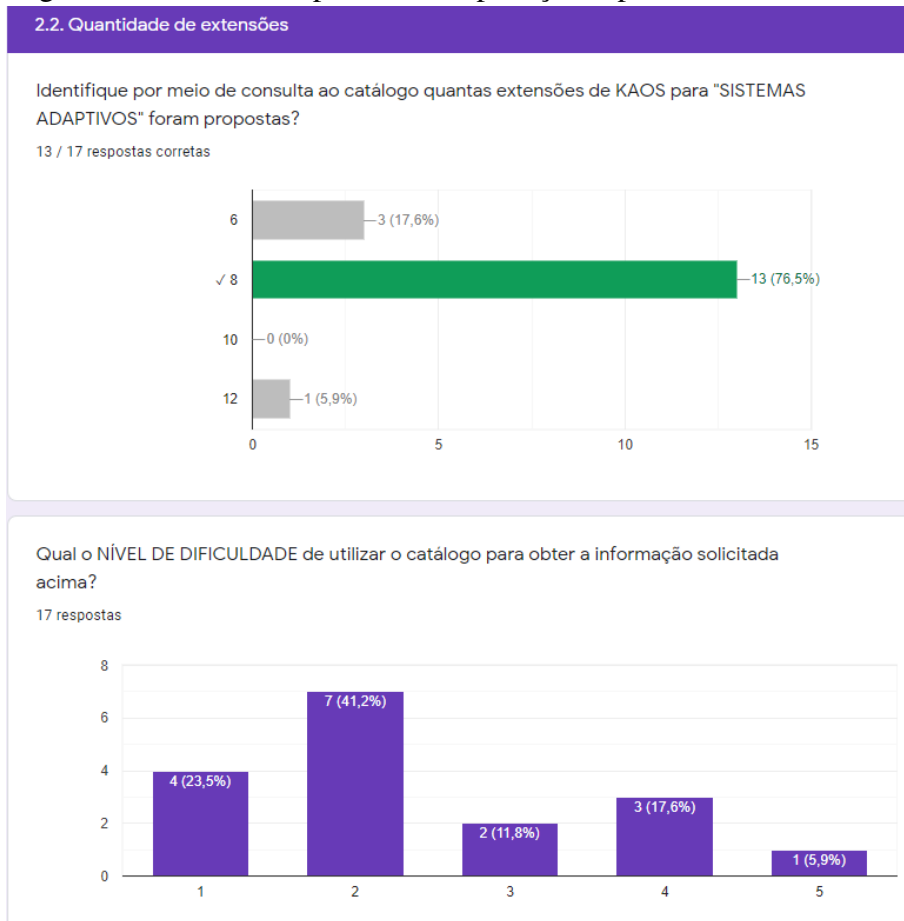
Figura 19 – Consultar Extensão



Fonte: Elaborado pelo Autor, (2022)

A Figura 20 mostra os resultados obtidos com a tarefa relacionada a quantidade de extensões de uma área de aplicação específica. Mais de 75% dos participantes respondeu de forma correta a essa questão. Para essa questão os participantes tiveram uma maior variação na percepção de dificuldade, mas ainda assim a maioria ainda concentrou as respostas em fácil ou muito fácil.

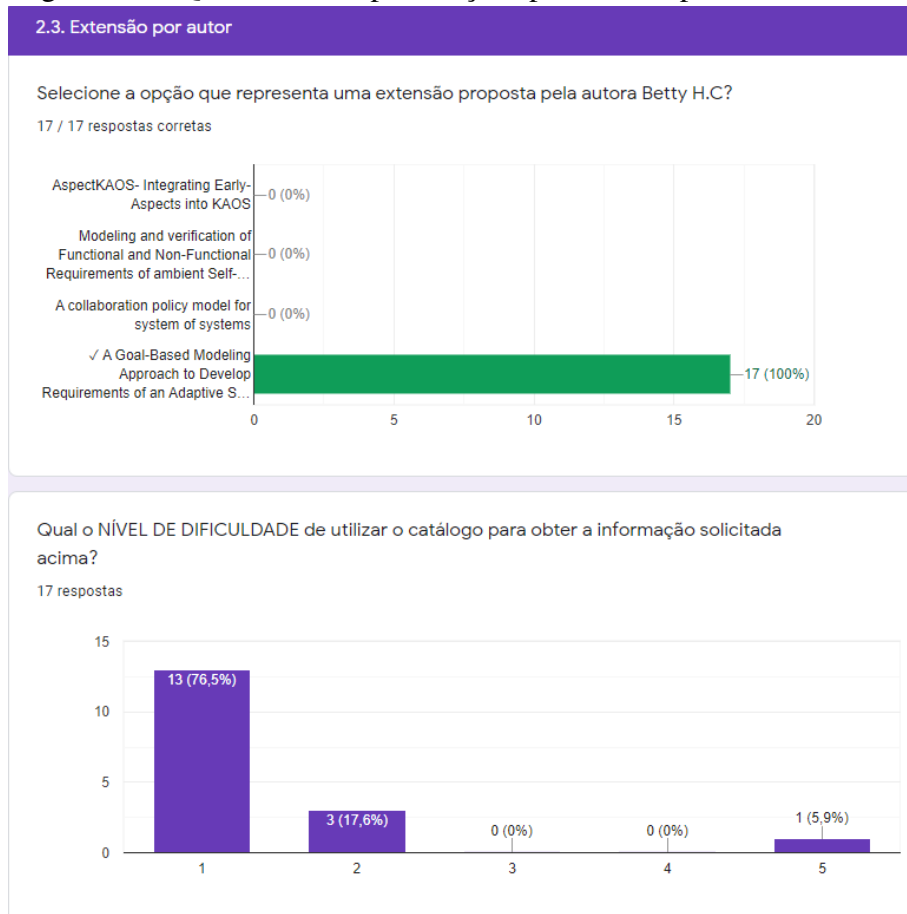
Figura 20 – Extensões por área de aplicação específica



Fonte: Elaborado pelo Autor, (2022)

A Figura 21 mostra os resultados obtidos com relação a tarefa de identificar a extensão proposta por um autor específico. Nesta tarefa 100% dos participantes respondeu de forma correta e pelo menos 94% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

Figura 21 – Quantidade de publicações por autor específico

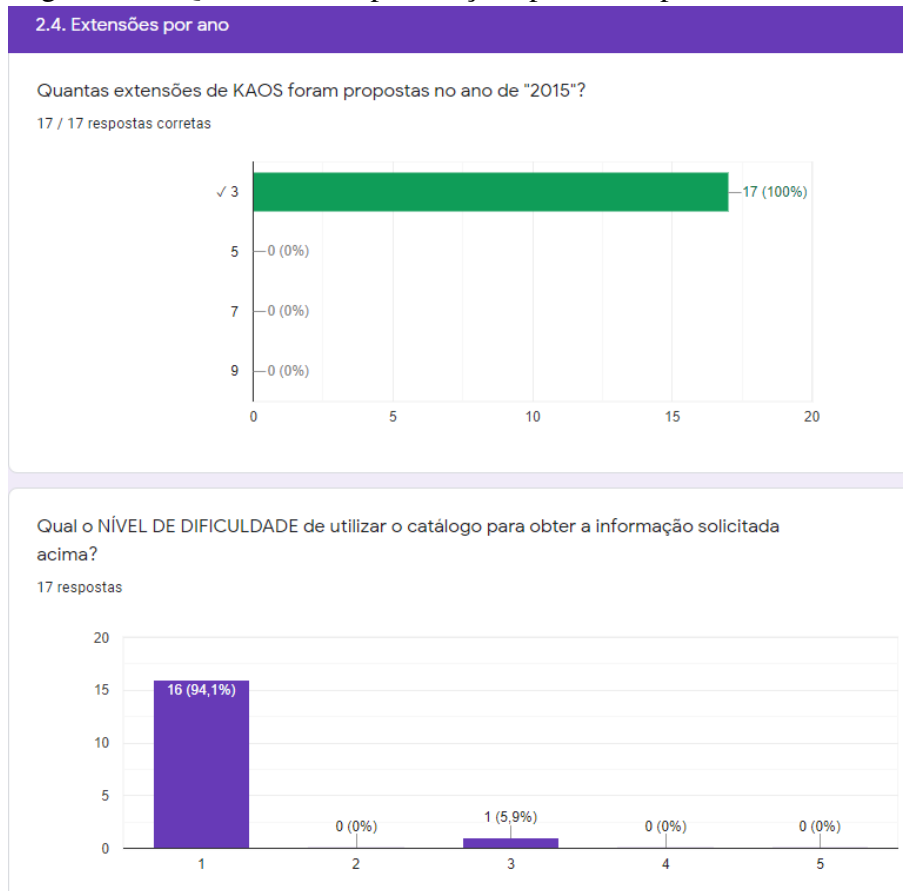


Fonte: Elaborado pelo Autor, (2022)

A Figura 22 mostra os resultados obtidos com relação a tarefa de identificar a quantidade de extensões de KAOS propostas em um ano específico. Para esta tarefa 100% dos participantes também respondeu de forma correta e pelo menos 94% julgou como muito fácil obter a resposta usando o catálogo.



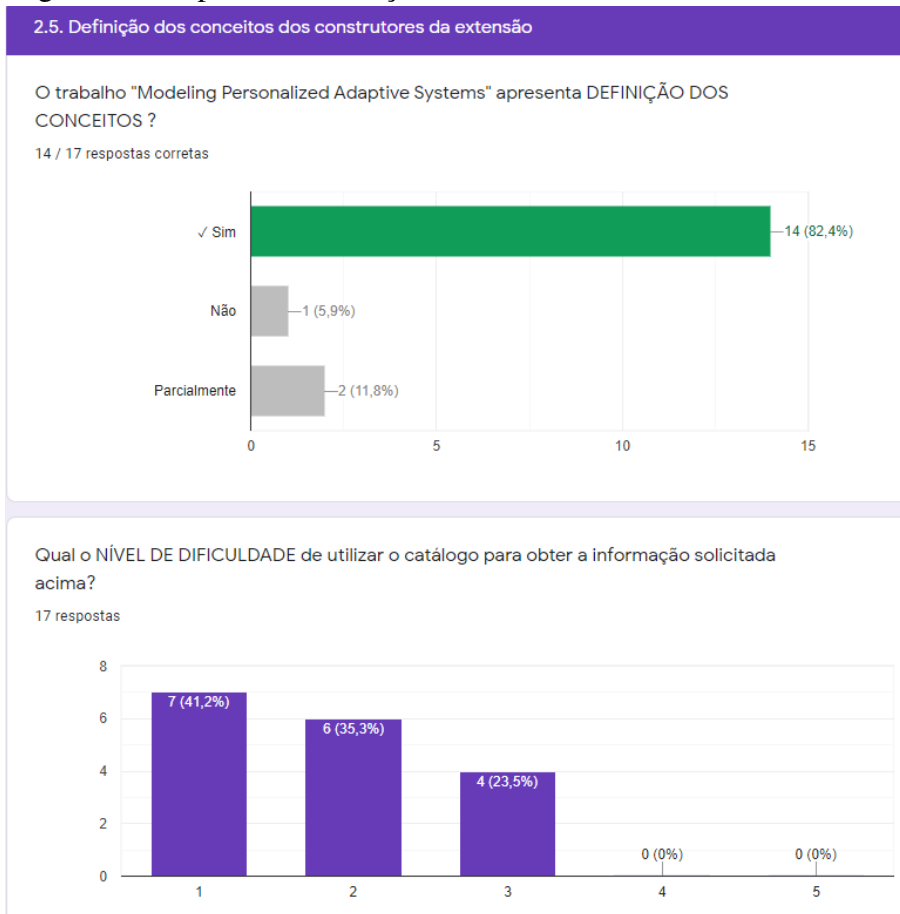
Figura 22 – Quantidade de publicações por ano específico



Fonte: Elaborado pelo Autor, (2022)

A Figura 23 mostra os resultados obtidos com relação a tarefa de identificar se uma extensão específica apresenta, não apresenta ou apresenta parcialmente a definição dos conceitos dos construtores. Mais de 80% dos participantes respondeu essa pergunta de forma correta e pelo menos 75% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

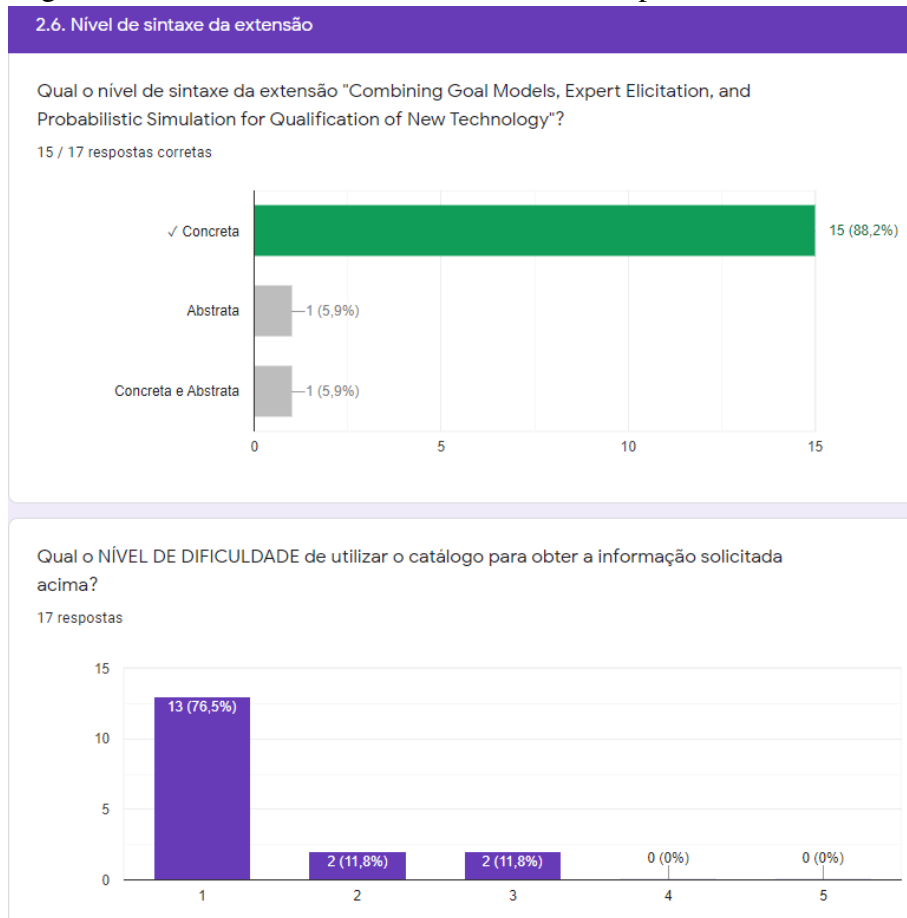
Figura 23 – Apresenta definição dos conceitos dos construtores



Fonte: Elaborado pelo Autor, (2022)

A respeito do nível de sintaxe de uma extensão específica, a Figura 24 mostra que aproximadamente 90% dos participantes respondeu de forma correta a tarefa que questionava qual o nível de sintaxe da extensão mencionada. Também aproximadamente 90% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

Figura 24 – Nível de sintaxe de uma extensão específica

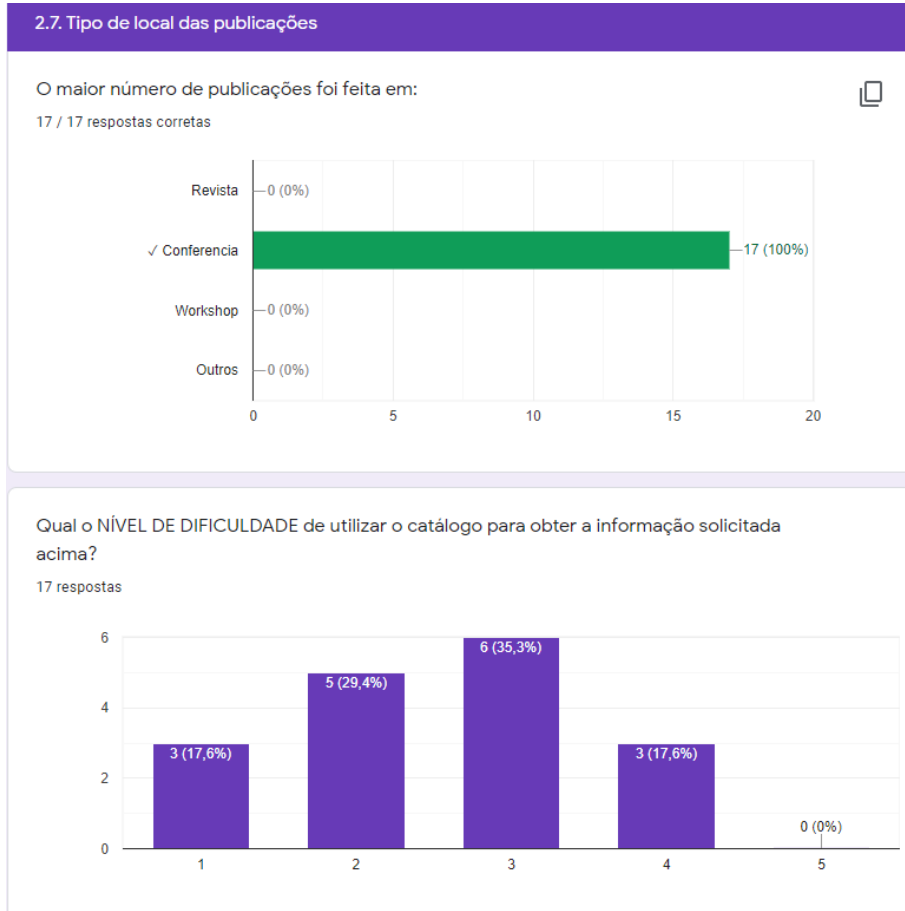


Fonte: Elaborado pelo Autor, (2022)

A respeito do tipo de local das publicações a Figura 25 mostra que 100% dos participantes tiveram sucesso ao usar o catálogo e perceber que a maior concentração de publicações foi feita em conferências. Essa foi uma tarefa em que alguns participantes tiveram um pouco mais de dificuldade.

Na parte final do questionário perguntamos a respeito de melhorias que poderiam ser adicionadas ao catálogo e exibir uma label com a quantidade de resultados por pesquisa foi uma das sugestões.

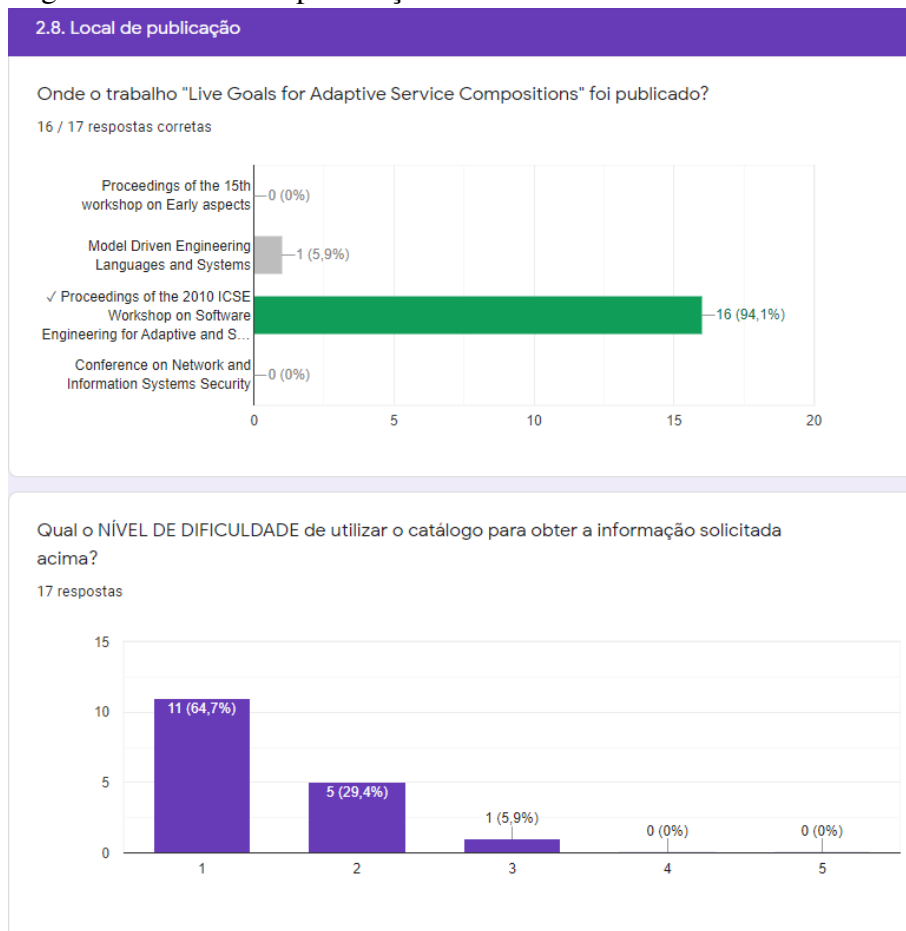
Figura 25 – Tipo de local de uma publicação de uma extensão



Fonte: Elaborado pelo Autor, (2022)

A Figura 26 mostra os resultados obtidos com relação a tarefa de identificar o local de publicação de uma extensão específica. Mais de 90% dos participantes respondeu essa pergunta de forma correta e mais de 90% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

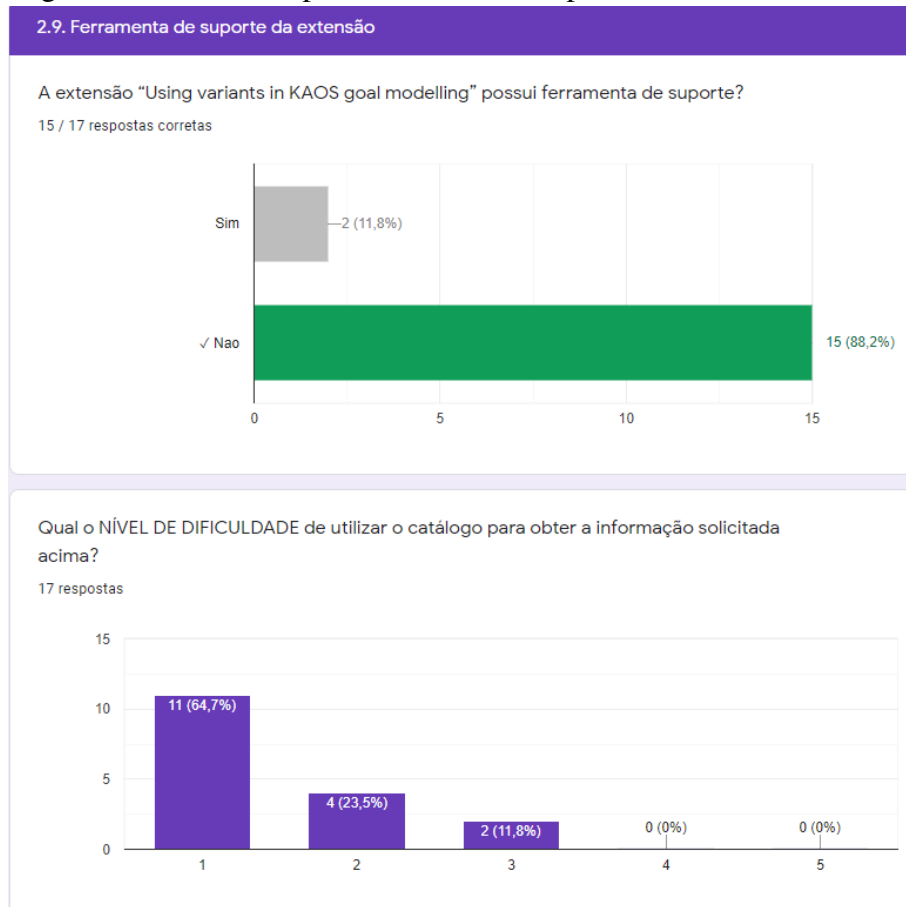
Figura 26 – Local de publicação de uma extensão



Fonte: Elaborado pelo Autor, (2022)

A Figura 27 mostra os resultados obtidos em relação a tarefa de identificar se uma extensão específica possui ferramenta suporte. Aproximadamente 90% dos participantes respondeu essa pergunta de forma correta e também aproximadamente 90% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

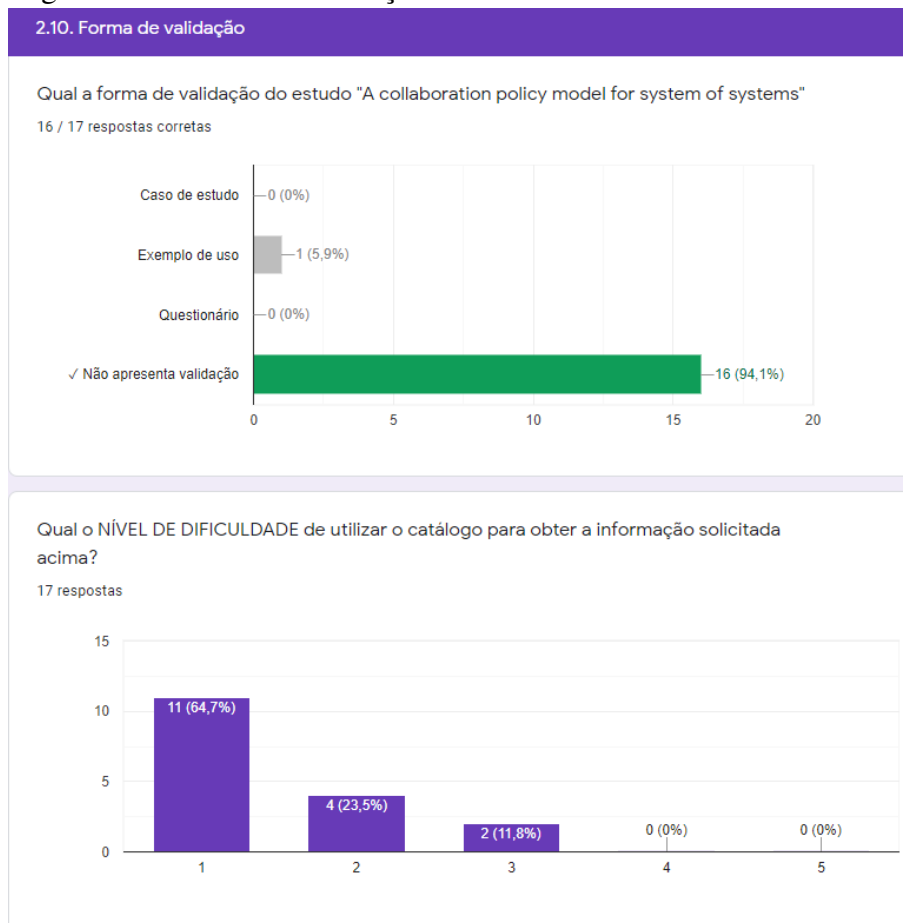
Figura 27 – Extensão possui ferramenta suporte



Fonte: Elaborado pelo Autor, (2022)

A Figura 28 mostra os resultados obtidos em relação a tarefa de identificar qual a forma de validação de uma extensão específica. Mais de 90% dos participantes respondeu essa pergunta de forma correta e aproximadamente 90% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

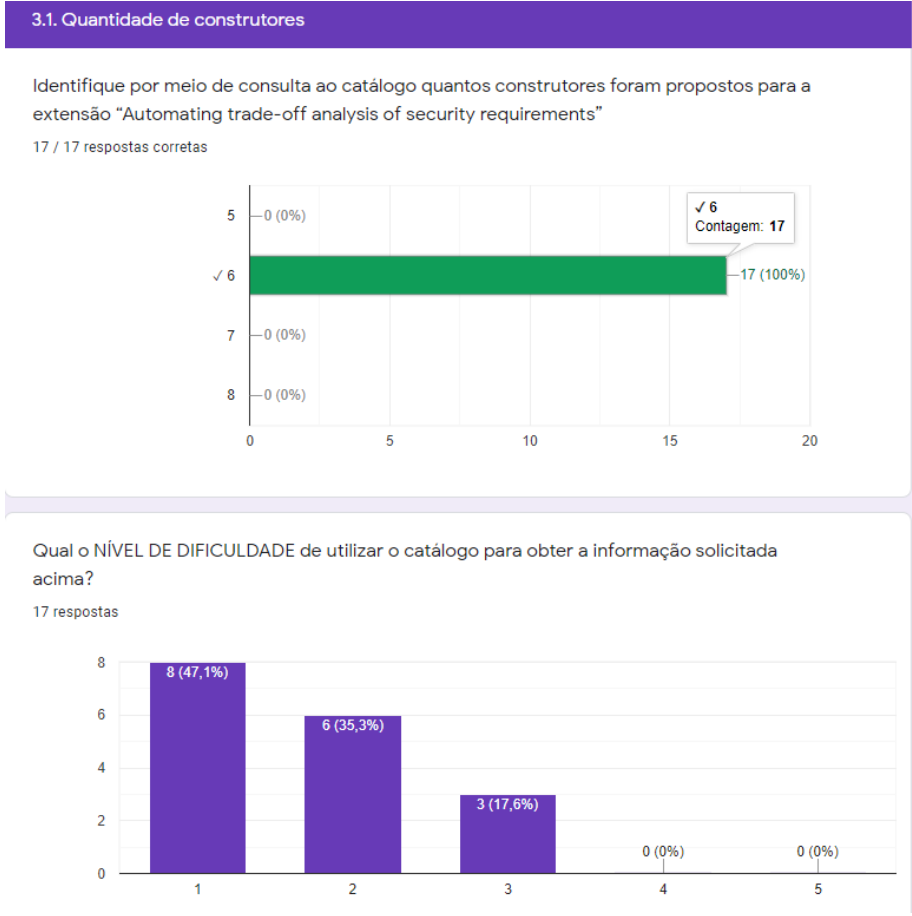
Figura 28 – Forma de validação de uma extensão



Fonte: Elaborado pelo Autor, (2022)

A Figura 29 mostra os resultados obtidos em relação a tarefa de identificar quantos construtores pertencem a uma extensão específica. 100% dos participantes respondeu essa pergunta de forma correta e mais de 80% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

Figura 29 – Quantidade de construtores propostos em uma extensão

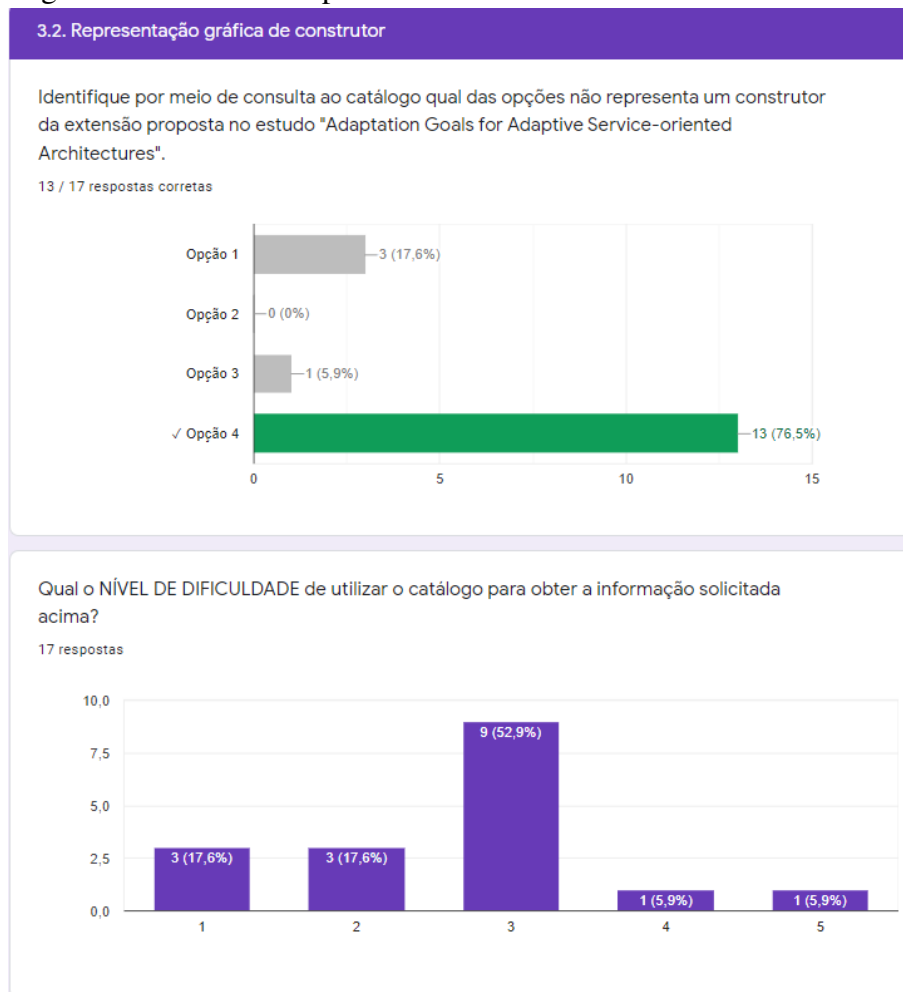


Fonte: Elaborado pelo Autor, (2022)

A Figura 30 mostra os resultados obtidos em relação a tarefa de identificar por através de exemplo visual qual construtor pertence a uma extensão específica. Esta foi uma tarefa que teve um pouco mais de dificuldade por parte de alguns participantes, mas pelo menos 75% obteve sucesso ao usar o catálogo para responder a pergunta de forma correta.



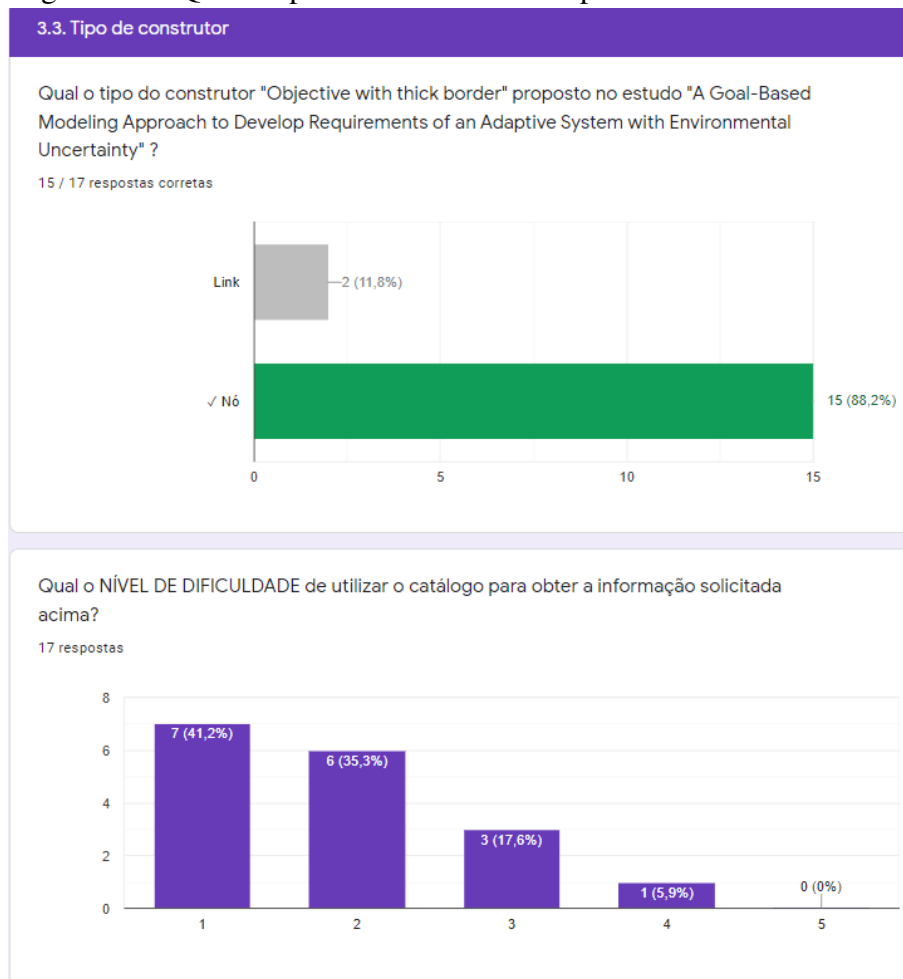
Figura 30 – Construtor pertence há uma extensão



Fonte: Elaborado pelo Autor, (2022)

A Figura 31 mostra os resultados obtidos em relação a tarefa de identificar qual o tipo de um construtor específico. Aproximadamente 90% dos participantes respondeu essa pergunta de forma correta e pelo menos 75% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

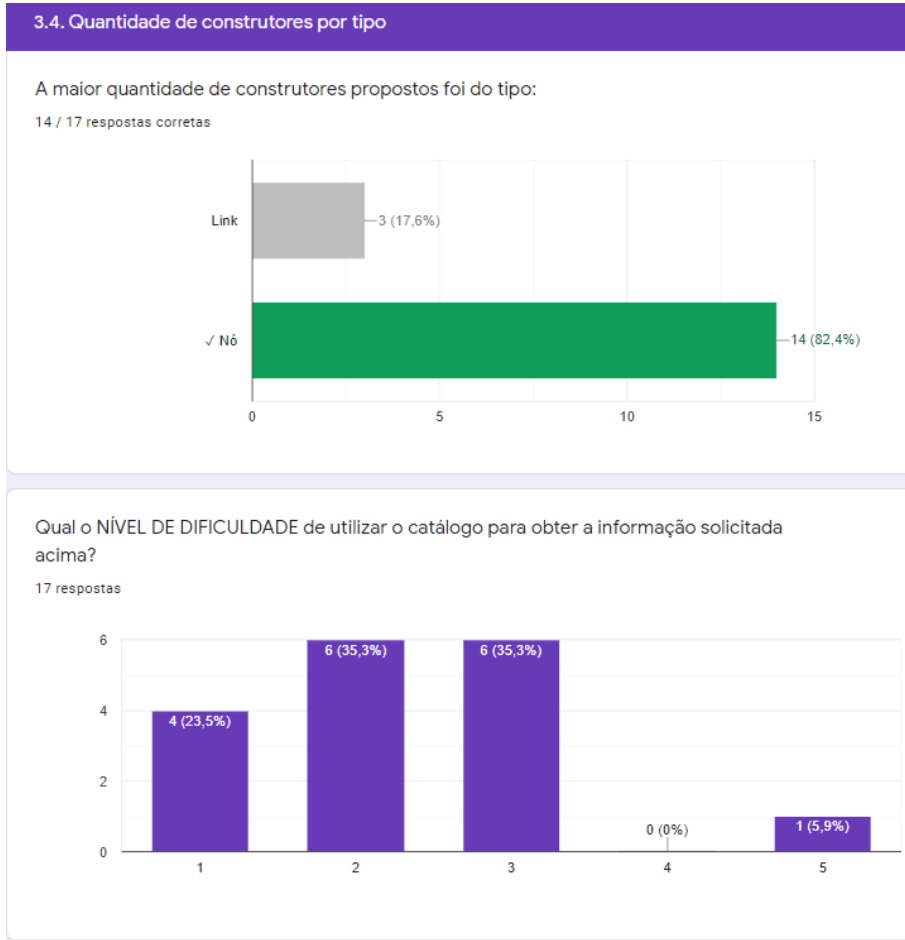
Figura 31 – Qual o tipo de um construtor específico



Fonte: Elaborado pelo Autor, (2022)

A Figura 32 mostra os resultados obtidos em relação a tarefa de identificar qual o tipo de construtor que mais foi proposto dentre as extensões. Esta também foi uma tarefa que teve um pouco mais de dificuldade por parte de alguns participantes, mas ainda assim pelo menos 80% obteve sucesso ao usar o catálogo para responder a pergunta de forma correta.

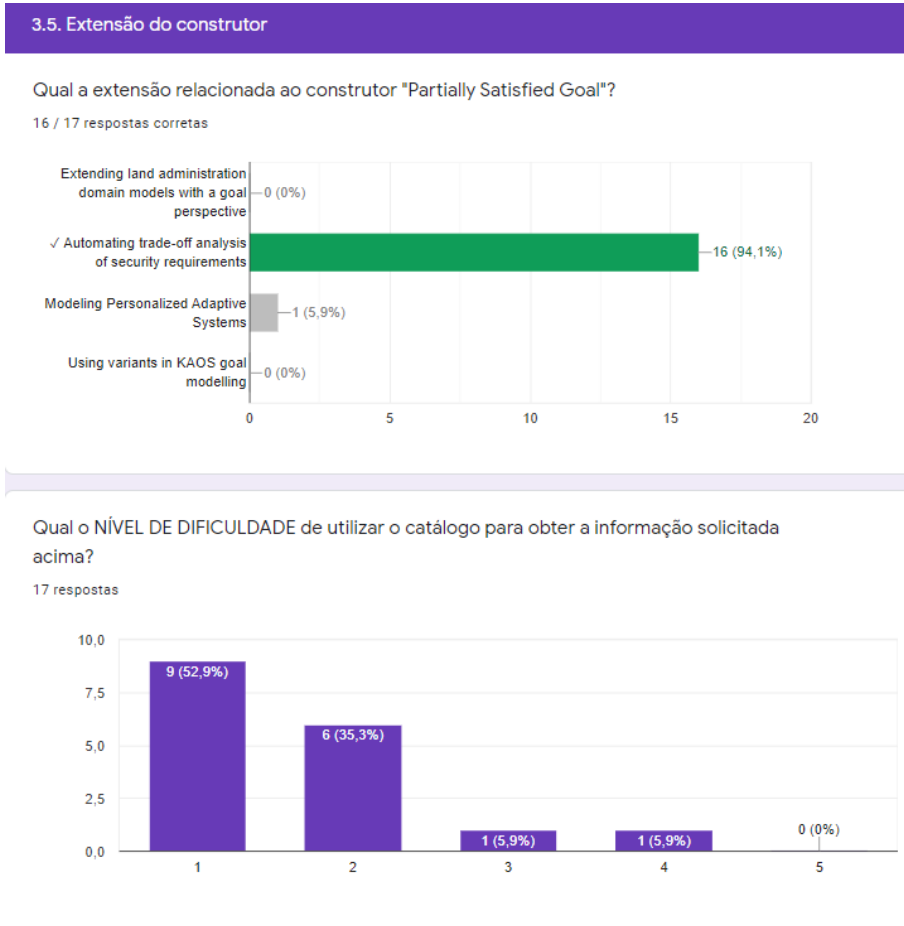
Figura 32 – Tipo de construtor mais proposto



Fonte: Elaborado pelo Autor, (2022)

A Figura 33 mostra os resultados obtidos em relação a tarefa de identificar qual a extensão relacionada a um construtor específico. Mais de 90% dos participantes respondeu essa pergunta de forma correta e aproximadamente 90% julgou como muito fácil ou fácil obter a resposta usando o catálogo.

Figura 33 – Extensão relacionada a um construtor específico



Fonte: Elaborado pelo Autor, (2022)

Na ultima parte do questionário, pedimos que os participantes respondessem questões relacionadas ao uso do catálogo de forma geral. Na primeira questão pedimos para que os participantes descrevessem de forma livre como foi a experiencia com o uso do catálogo. A Figura 34 mostra na integra os comentários dos participantes.

Figura 34 – Opinião dos participantes experiência de utilizar o catálogo



Fonte: Elaborado pelo Autor, (2022)

Nas questões 4.2, 4.3 e 4.4, os participantes selecionaram em uma escala que vai de 0 (discordo totalmente) a 5 (concordo totalmente) quais as suas impressões em relação a uma afirmação sobre o catálogo.

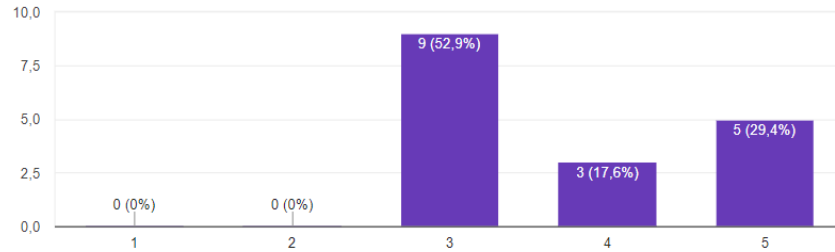
Nas figuras 35, 36 e 37 é possível ver a opinião dos participantes em relação as questões supracitadas.

Figura 35 – Impressões sobre layout do catálogo

4.2. Selecione a opção que representa sua opinião sobre a seguinte afirmação: O layout do catálogo é amigável



17 respostas



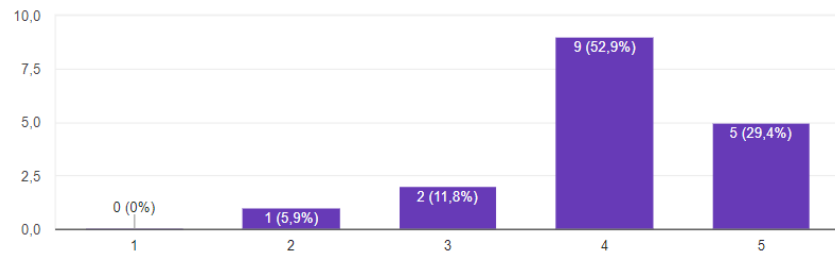
Fonte: Elaborado pelo Autor, (2022)

Figura 36 – Impressões sobre a complexidade das tarefas

4.3. Selecione a opção que representa sua opinião sobre a seguinte afirmação: As funcionalidades podem ser executadas de forma intuitiva



17 respostas



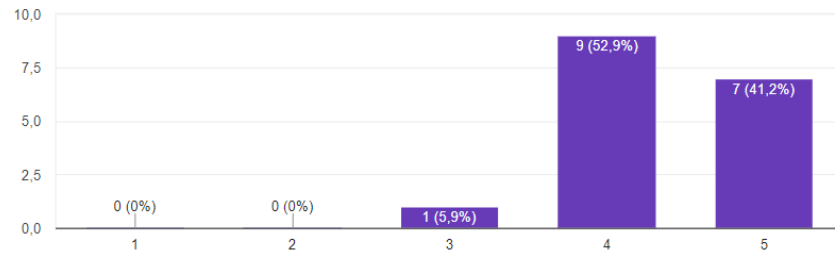
Fonte: Elaborado pelo Autor, (2022)

Figura 37 – Impressões sobre velocidade e corretude das consultas

4.4. Selecione a opção que representa sua opinião sobre a seguinte afirmação: O catálogo foi útil para identificação rápida e correta das consultas realizadas



17 respostas



Fonte: Elaborado pelo Autor, (2022)

## 6.4 Discussões

Analisando os dados obtidos é possível identificar que: aproximadamente 47,06% (8/17) dos participantes tiveram uma taxa de acerto 100%; 23,53% (4/17) dos participantes teve uma taxa de acerto  $< 100\%$  e  $\geq 90\%$ ; 23,53% (4/17) dos participantes tiveram uma taxa de acerto  $< 90\%$  e  $\geq 80\%$ . Apenas 1 dos participantes (5,88) atingiu um percentual de acertos inferior a 80%.

Fazendo ainda uma análise sobre cada questão é possível ver que aproximadamente quatro questões (4/15) tiveram uma taxa de acerto de 100%; outras quatro questões (4/15) tiveram uma taxa de acerto  $< 100\%$  e  $\geq 90\%$ ; Outras cinco questões (5/15) tiveram uma taxa de acerto  $< 90\%$  e  $\geq 80\%$ ; Apenas duas questões (2/15) atingiram um percentual de acertos inferior a 80%.

Dentre as questões com um menor percentual estão as questões 2.2 como mostra a Figura 20 e a questão 3.2 como mostra a Figura 30. Levando em consideração as sugestões de melhorias propostas pelos participantes percebe-se que entre elas está a sugestão de mostrar a quantidade de resultados obtidos ao aplicar um filtro na lista de extensões o que ajudaria na tarefa proposta para responder a questão 2.2.

Sobre a percepção dos participantes da dificuldade de executar as ações, os resultados obtidos mostram que em 73,33% das questões (11/15) os participantes julgaram como muito fácil ou fácil obter a resposta usando o catálogo.

De uma forma geral, como mostra na Figura 34 os participantes do estudo de avaliação descreveram a experiência de uso do catálogo como positiva. Dentre os principais características apontadas pelos mesmos estão: simples, intuitiva, rápida, fluida e eficiente.

A Figura 35 apresenta as impressões sobre o layout do catálogo onde menos de 50% dos participantes concordaram com a afirmação que diz que o layout é amigável. Esse dado demonstra um ponto a ser analisado e aprimorado por próximos trabalhos.

A Figura 36 exhibe as impressões sobre a complexidade de executar algumas tarefas no catálogo e mostrou que pelo menos 80% dos participantes concorda com a afirmação que diz que as funcionalidades podem ser executadas de forma intuitiva.

A Figura 37 mostra as impressões sobre a velocidade de uso do catálogo nas consultas e também mostrou um bom percentual onde pelo menos 90% dos participantes concordou com a afirmação de que o catálogo foi útil na identificação rápida e correta das extensões.

Todos os resultados do questionário estão em uma planilha disponível no link <http://bit.do/fTnHA>.

## **6.5 Ameaças a validade**

Esta seção discute sobre as ameaças a validade do instrumento da pesquisa, ou seja, se o mesmo vai medir o que realmente deseja medir na pesquisa. Para (KITCHENHAM; PFLEEGER, 2002) 4 (quatro) aspectos devem ser considerados: Validade de Face, Validade de Conteúdo, Validade de Conclusão, Validade de Construção.

### **6.5.1 Validade de face**

Pode ser entendida como uma revisão superficial dos itens do instrumento por pessoas inexperientes. O questionário feito no *Google Forms* foi apresentado para pessoas que não tinham conhecimento no assunto avaliado pelo *survey*, com o objetivo de revisar a estrutura, o design e a objetividade das tarefas do questionário. Também foi executado um teste piloto, com o intuito de avaliar a compreensão das questões e se todas as questões poderiam de fato ser respondidas de forma correta usando a plataforma do catálogo. Mesmo sem conhecimentos prévios sobre o assunto, os participantes conseguiram um elevado percentual de acertos as questões.

### **6.5.2 Validade de conteúdo**

Trata-se de uma avaliação subjetiva de como o instrumento parece adequado a um grupo de pessoas com conhecimento sobre o assunto. Realizamos outro piloto envolvendo duas pessoas sendo uma delas um especialista em engenharia de requisitos e o outro um estudante do Programa de Mestrado Acadêmico em Computação da UFC. O objetivo foi testar a compreensão dos participantes sobre a pesquisa e assegurar que ela inclui todas as demandas necessárias para a validação. O feedback recebido neste piloto apresentou alguns pontos de melhoria na estrutura do catálogo e a necessidade de inclusão de perguntas qualitativas a respeito da opinião dos participantes sobre o uso do catálogo de forma geral.

### **6.5.3 Validade de conclusão**

Diz respeito a habilidade de alcançar o objetivo correto sobre os dados coletados, utilizando testes estatísticos, e quão confiáveis são as medidas e esses dados. Devido a um não tão alto de participantes neste estudo, não poderíamos fazer inferências estatísticas sobre os dados o que gerou uma ameaça a validade deste estudo.



#### **6.5.4 Validade de construção**

É a observação de como o instrumento de pesquisa porta-se quando está em uso. Os dados obtidos podem ser convergentes ou divergentes a depender de problemas que possam surgir, sejam por falhas do pesquisador ou dos participantes, por exemplo:

- Os participantes podem basear seu comportamento em suposições.
- O ser humano geralmente tenta parecer melhor do que é quando está sendo avaliado
- Os pesquisadores podem projetar o experimento pensando nos resultados que esperam (viés)

Para tentar mitigar essa ameaça buscamos elaborar questões objetivas, que foram validadas nos testes piloto que foram realizados. Nos resultado deste estudo que são melhores detalhados na seção 6.3, vimos que os dados convergiram na direção a um alto percentual de sucesso ao utilizar o catálogo para responder as questões do nosso instrumento de pesquisa.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

KAOS é uma linguagem de modelagem cujo foco principal é representar explicitamente os objetivos dos sistemas, os conflitos e obstáculos entre esses objetivos, quais os objetos responsáveis por satisfazê-los e quais as consequências do resultado da interação entre os mesmos.

Desde a sua criação na década de 90, ela vem sendo amplamente utilizada e estendida. A identificação de extensões de forma manual e seus construtores pode ser uma tarefa complexa que requer grande quantidade de tempo e tem uma propensão a não ser exitosa quando executada sem o apoio adequado.

Neste trabalho apresentamos um catálogo de extensões de KAOS com o intuito de otimizar a identificação e visualização das extensões e de seus construtores. O catálogo está online e nele é possível ver a lista de todas as extensões encontradas na RSL e acessar as informações detalhadas de cada um incluindo um link para acessar o local de publicação. A lista dos construtores que pertencem aquela extensão também é exibida na página de detalhes. A ferramenta permite a busca de extensões por autor, título e permite filtrar por área de aplicação, ano de publicação, local de publicação ou se ela tem ferramenta de suporte.

Ainda neste trabalho o catálogo foi avaliado por meio de um estudo exploratório baseado em *survey* que teve como objetivo analisar qual o efeito do uso do catálogo na identificação das extensões de KAOS e seus construtores. A avaliação procurava mensurar a correteza da identificação de informações de extensões por meio do catálogo de KAOS e o esforço envolvido.

Como resultados da avaliação podemos identificar que pelo menos 90% dos participantes obteve sucesso ao responder pelo menos 12 (doze) das 15 (quinze) perguntas propostas no questionário. Ainda aliado a esse bom desempenho era importante analisar o grau de dificuldade que os participantes tiveram para conseguir realizar essas tarefas e ao questioná-los sobre isso, vimos que em aproximadamente 75% das questões os participantes julgaram como muito fácil ou fácil obter a resposta usando o catálogo.

Ao fim do questionário perguntamos sobre sugestões de melhorias e algumas foram apontadas. Dentre as sugestões estão a criação de um design mais convidativo, a inclusão de novos filtros de pesquisa, a adição de mais labels a inputs. De forma geral as sugestões eram sobre melhorar a interface de usuário e deixá-la mais atrativa.

Como trabalhos futuros, identificamos a possibilidade de um estudo que analise a usabilidade do catálogo, como por exemplo uma avaliação heurística que é uma técnica de

inspeção de usabilidade executada por examinadores que seguem um conjunto de princípios de usabilidade, as heurísticas, e avaliam todos os elementos de interface com o usuário, com o objetivo de encontrar falhas de usabilidade.

Pretendemos também coletar mais respostas ao questionário de avaliação de modo a trazer uma maior abrangência na população.

Acreditamos ainda que um experimento baseado nos princípios apresentados por Wohlin *et al.* (2012) também seria importante para ratificar a importância do uso do catálogo na identificação e visualização das extensões de KAOS.

O experimento é criado para testar uma teoria ou hipótese. No projeto do experimento, um número de variáveis sobre quais pesquisadores têm controle são observados e uma pergunta de pesquisa é definida. Para este experimento poderia ser por exemplo: qual é o efeito do uso do catálogo para identificar extensões e seus construtores?

A principal hipótese é a hipótese nula que afirma que não há diferença entre usar ou não a ferramenta proposta neste trabalho. Portanto, o estudo tenta rejeitar esta hipótese.

A fim de identificar os efeitos da utilização da ferramenta proposta, o objeto de controle é a geração do objeto experimental sem o método proposto. Desta forma, os participantes serão divididos em dois grupos: Grupo A e Grupo B. No Grupo A estarão os participantes que utilizarão a ferramenta proposta (catálogo de extensões do KAOS), enquanto no Grupo B estarão os participantes que não usarão o método proposto (sem utilização do catálogo de extensões do KAOS).

Com os resultados dos 2 (dois) grupos poderemos comparar e discutir se houve um ganho de performance na busca das extensões de KAOS.

## REFERÊNCIAS

- BRAMBILLA, M.; CABOT, J.; WIMMER, M. Model-driven software engineering in practice. **Synthesis Lectures on Software Engineering**, [S.l.], v. 1, n. 1, p. 1–182, 2012.
- CARVALHO, R.; ANDRADE. **Correlate lead**: process and catalog of non-functional requirements correlations in ubicomp and iot systems 2017. 164 f. Tese (Doutorado em Ciência da Computação) — Centro de Ciências e Tecnologia, Departamento de Computação, Programa de Mestrado e Doutorado em Ciência da Computação. Universidade Federal do Ceará, Fortaleza, 2019.
- CARVALHO, R.; ANDRADE, R.; OLIVEIRA, K.; KOLSKI, C. Catalog of invisibility requirements for ubicomp and iot applications. In: IEEE. **2018 IEEE 26th International Requirements Engineering Conference (RE)**. [S.l.], 2018. p. 88–99.
- CHUNG, L.; NIXON, B. A.; YU, E.; MYLOPOULOS, J. **Non-functional requirements in software engineering**. [S.l.]: Springer Science & Business Media, 2012. v. 5.
- COSTEN, F. **Development of Three-dimensional Model of Avocado**. Tese (Doutorado) — The University of Manchester, 2019.
- CYSNEIROS, L. M. Evaluating the effectiveness of using catalogues to elicit non-functional requirements. In: **WER**. [S.l.: s.n.], 2007. p. 107–115.
- DARDENNE, A.; LAMSWEERDE, A. V.; FICKAS, S. Goal-directed requirements acquisition. **Science of computer programming**, Elsevier, v. 20, n. 1-2, p. 3–50, 1993.
- FATIMA, M. Kaos: A goal oriented requirement engineering approach. **International Journal for Innovative Research in Science & Technology**, v. 1, n. 10, p. 133–135, 2015.
- FAVERI, C. D.; MOREIRA, A.; AMARAL, V. Goal-driven deception tactics design. In: IEEE. **2016 IEEE 27th International Symposium on Software Reliability Engineering (ISSRE)**. [S.l.], 2016. p. 264–275.
- FRANCH, X.; MATÉ, A.; TRUJILLO, J. C.; CARES, C. On the joint use of i with other modelling frameworks: A vision paper. In: IEEE. **2011 IEEE 19th International Requirements Engineering Conference**. [S.l.], 2011. p. 133–142.
- GONCALVES, E.; CASTRO, J.; ARAUJO, J.; HEINECK, T. A systematic literature review of istar extensions. **Journal of Systems and Software**, Elsevier, v. 137, p. 1–33, 2018a.
- GONÇALVES, E.; HEINECK, T.; ARAÚJO, J.; CASTRO, J. A catalogue of istar extensions. In: **WER**. [S.l.: s.n.], 2018b.
- HEAVEN, W.; FINKELSTEIN, A. Uml profile to support requirements engineering with kaos. **IEE Proceedings-Software**, IET, v. 151, n. 1, p. 10–27, 2004.
- KELLY, S.; TOLVANEN, J.-P. **Domain-specific modeling**: enabling full code generation. [S.l.]: John Wiley & Sons, 2008.
- KITCHENHAM, B.; PFLEEGER, S. L. Principles of survey research part 4: questionnaire evaluation. **ACM SIGSOFT Software Engineering Notes**, ACM New York, NY, USA, v. 27, n. 3, p. 20–23, 2002.

- KOTONYA, G.; SOMMERVILLE, I. **Requirements engineering: processes and techniques**. [S.l.]: Wiley Publishing, 1998.
- LAMDDI, M. A. Developing dependability requirements engineering for secure and safe information systems with knowledge acquisition for automated specification. **Journal of Software Engineering and Applications**, Scientific Research Publishing, v. 10, n. 02, p. 211, 2017.
- MUSSBACHER, G.; AMYOT, D.; BREU, R.; BRUEL, J.-M.; CHENG, B. H.; COLLET, P.; COMBEMALE, B.; FRANCE, R. B.; HELDAL, R.; HILL, J. *et al.* The relevance of model-driven engineering thirty years from now. In: SPRINGER. **International Conference on Model Driven Engineering Languages and Systems**. [S.l.], 2014. p. 183–200.
- MYLOPOULOS, L. C. J.; YU, E. **From object-oriented to goal-oriented Requirements analysis**. [S.l.: s.n.], 1999.
- PACHECO, C. L.; GARCIA, I. A.; CALVO-MANZANO, J. A.; ARCILLA, M. A proposed model for reuse of software requirements in requirements catalog. **Journal of Software: Evolution and Process**, Wiley Online Library, v. 27, n. 1, p. 1–21, 2015.
- PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016.
- PRESSMAN, R. S. **Software engineering: a practitioner's approach**. [S.l.]: Palgrave Macmillan, 2005.
- RESPECT, I. **A KAOS Tutorial. Objectiver**. [S.l.]: línea, 2007.
- SEMMAK, F.; GNAHO, C.; LALEAU, R. Extended kaos to support variability for goal oriented requirements reuse. In: **MoDISE-EUS**. [S.l.: s.n.], 2008. p. 22–33.
- SILVA, R. C. d. *et al.* **Uma abordagem para o reuso de requisitos baseada em padrões e rastreabilidade**. [S.l.]: Universidade do Vale do Itajaí, 2011.
- SOMMERVILLE, I. **Software engineering**. [S.l.]: New York: Addison-Wesley, 2011.
- SOUZA, S. C. **Uma revisão sistemática de literatura das extensões de kaos (knowledge acquisition in automated specification)**. [S.l.]: Trabalho de conclusão de Curso, Universidade Federal do Ceará, 2019.
- WOHLIN, C.; RUNESON, P.; HÖST, M.; OHLSSON, M. C.; REGNELL, B.; WESSLÉN, A. **Experimentation in software engineering**. [S.l.]: Springer Science & Business Media, 2012.
- YIN, R. K. **Estudo de Caso: Planejamento e métodos**. [S.l.]: Bookman editora, 2015.

## APÊNDICE A – QUESTIONÁRIO DE AVALIAÇÃO DO CATÁLOGO

### Caracterização do perfil do participante

1. Qual seu curso da UFC Quixadá?
  - a) Ciência da Computação
  - b) Design Digital
  - c) Engenharia de Software
  - d) Engenharia da Computação
  - e) Redes de Computadores
  - f) Design Digital
  
2. Qual semestre no curso você está?
  - a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
  - f) 6
  - g) 7
  - h) 8
  - i) 9
  - j) 10
  
3. Numa escala de 0 (sem conhecimento) até 5 (especialista), qual seu nível de conhecimento em Engenharia Requisitos?
  - a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

4. Numa escala de 0 (sem conhecimento) até 5 (especialista), qual seu nível de conhecimento em Modelagem de Sistemas?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

**Perguntas relacionadas às extensões do catálogo**

1. O trabalho "A comparison of goal-oriented approaches to model software product lines variability" é uma extensão de KAOS?
- a) Sim
  - b) Não
2. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
3. Quantas extensões de KAOS para "SISTEMAS ADAPTIVOS" foram propostas?
- a) 6
  - b) 8
  - c) 10
  - d) 12
4. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

5. Que opção representa uma extensão proposta pela autora Betty H.C?
- a) AspectKAOS- Integrating Early-Aspects into KAOS
  - b) Modeling and verification of Functional and Non-Functional Requirements of ambient Self-Adaptive Systems
  - c) A collaboration policy model for system of systems
  - d) A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty
6. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
7. Quantas extensões de KAOS foram propostas no ano de "2015"?
- a) 3
  - b) 5
  - c) 7
  - d) 9
8. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
9. O trabalho "Modeling Personalized Adaptive Systems" apresenta DEFINIÇÃO DOS CONCEITOS ?
- a) Sim
  - b) Não
  - c) Parcialmente



10. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
11. Qual o nível de sintaxe da extensão "Combining Goal Models, Expert Elicitation, and Probabilistic Simulation for Qualification of New Technology"?
- a) Concreta
  - b) Abstrata
  - c) Concreta e Abstrata
12. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
13. Onde foi feito o maior número de publicações?
- a) Revista Científica
  - b) Conferência
  - c) Workshop
  - d) Outros
14. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

15. Onde o trabalho "Live Goals for Adaptive Service Compositions" foi publicado?
- a) Proceedings of the 15th workshop on Early aspects
  - b) Model Driven Engineering Languages and Systems
  - c) Proceedings of the 2010 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems
  - d) Conference on Network and Information Systems Security
16. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
17. A extensão "Using variants in KAOS goal modelling" possui ferramenta de suporte?
- a) Sim
  - b) Não
18. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
19. Qual a forma de validação do estudo "A collaboration policy model for system of systems"?
- a) Caso de estudo
  - b) Exemplo de uso
  - c) Questionário
  - d) Não apresenta validação

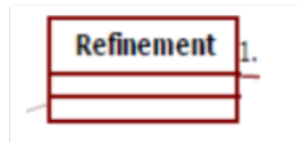
20. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

**Perguntas relacionadas aos construtores do catálogo**

1. Quantos construtores foram propostos para a extensão “Automating trade-off analysis of security requirements”?
- a) 5
  - b) 6
  - c) 7
  - d) 8
2. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

3. Qual das opções não representa um construtor da extensão proposta no estudo "Adaptation Goals for Adaptive Service-oriented Architectures"

a) -



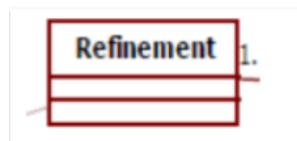
b) -



c) -



d) -



4. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?

- a) 1
- b) 2
- c) 3
- d) 4
- e) 5

5. Qual o tipo do construtor "Objective with thick border" proposto no estudo "A Goal-Based Modeling Approach to Develop Requirements of an Adaptive System with Environmental Uncertainty"?
- a) Link
  - b) Nó
6. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
7. A maior quantidade de construtores propostos foi de que tipo?
- a) Link
  - b) Nó
8. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

9. Qual a extensão relacionada ao construtor "Partially Satisfied Goal"?
- a) Extending land administration domain models with a goal perspective
  - b) Automating trade-off analysis of security requirements
  - c) Modeling Personalized Adaptive Systems
  - d) Using variants in KAOS goal modelling
10. Numa escala de 0 (muito fácil) até 5 (muito difícil), qual o NÍVEL DE DIFICULDADE de utilizar o catálogo para obter a informação solicitada na questão anterior?
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

**Perguntas relacionadas ao uso do catálogo de forma geral**

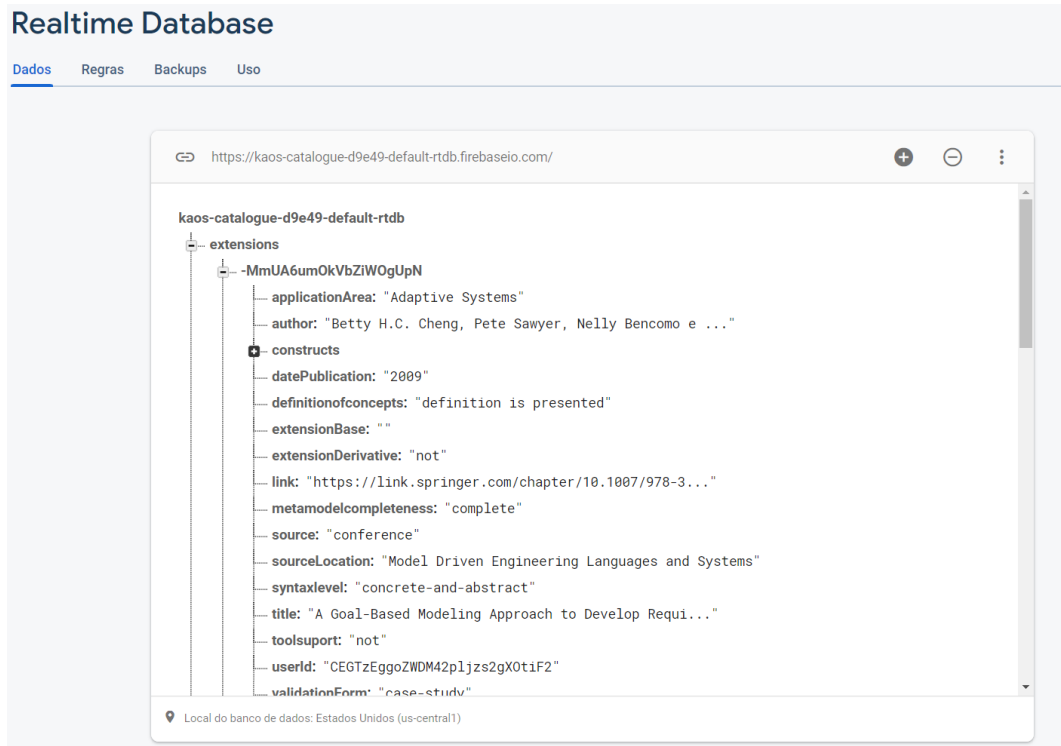
1. Como foi a experiência de utilizar o catálogo?
2. Numa escala de 0 (discordo fortemente) até 5 (concordo fortemente), selecione a opção que representa sua opinião sobre a seguinte afirmação: O layout do catálogo é amigável.
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
3. Numa escala de 0 (discordo fortemente) até 5 (concordo fortemente), selecione a opção que representa sua opinião sobre a seguinte afirmação: As funcionalidades podem ser executadas de forma intuitiva
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5

4. Numa escala de 0 (discordo fortemente) até 5 (concordo fortemente), selecione a opção que representa sua opinião sobre a seguinte afirmação: O catálogo foi útil para identificação rápida e correta das consultas realizadas.
- a) 1
  - b) 2
  - c) 3
  - d) 4
  - e) 5
5. Tem alguma sugestão para melhoria do catálogo?

## APÊNDICE B – TECNOLOGIAS UTILIZADAS

A Figura 38 mostra o console do *firebase* e como as informações ficam estruturadas em um JSON.

Figura 38 – Console do Realtime Database

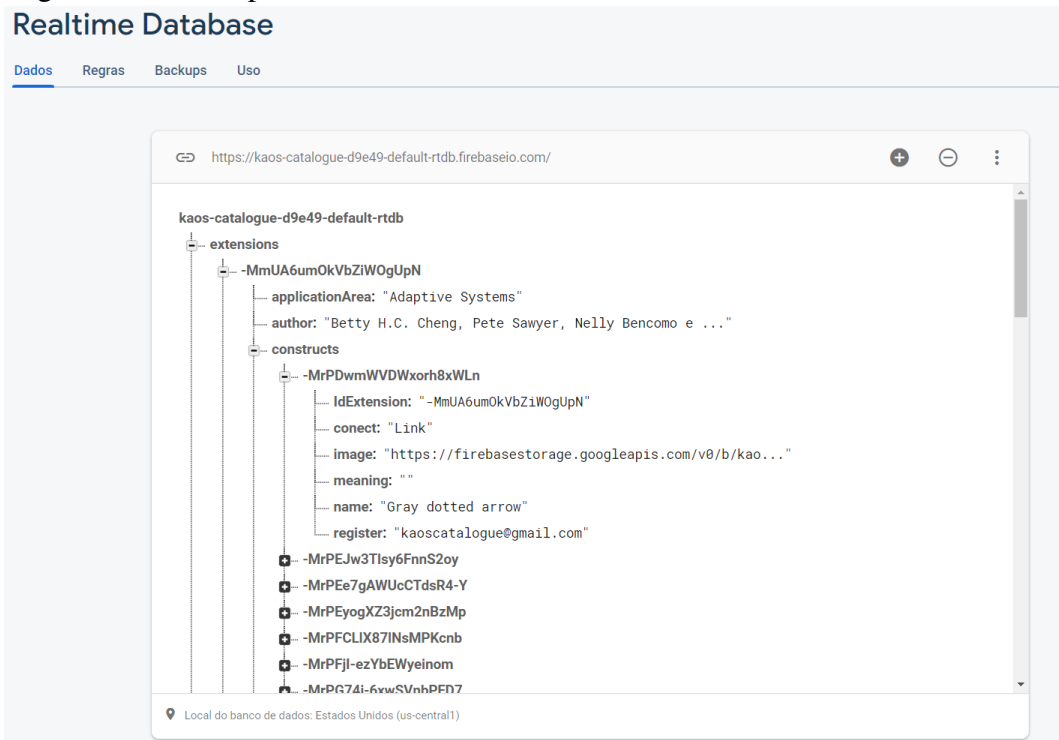


Fonte: Elaborado pelo Autor, (2022)

A Figura 39 mostra como as informações dos construtores ficam armazenadas dentro do JSON.



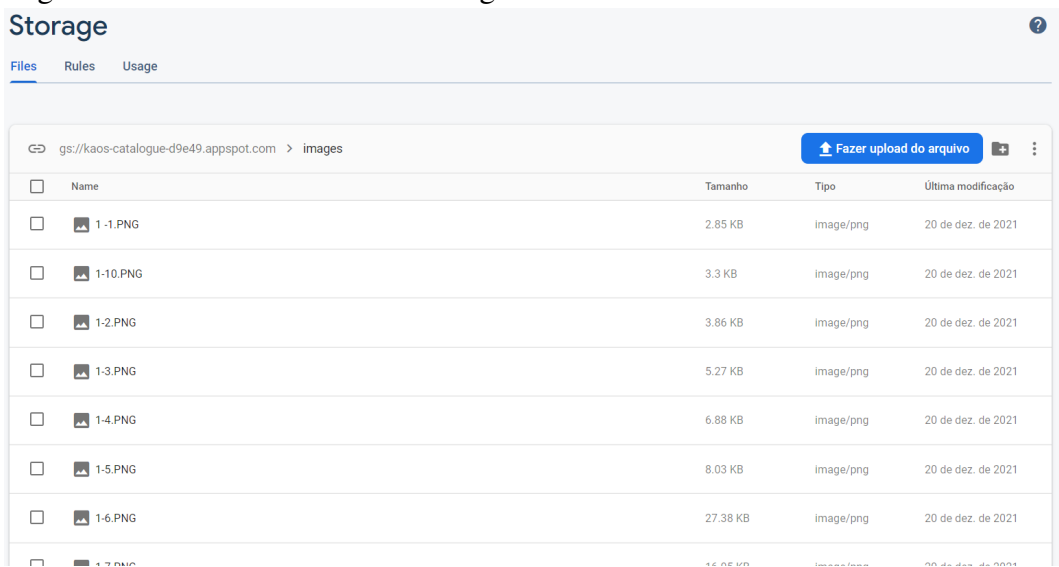
Figura 39 – Exemplo de lista dos construtores dentro do JSON de uma extensão Realtime Database



Fonte: Elaborado pelo Autor, (2022)

A Figura 40 mostra o console do *O Cloud Storage*.

Figura 40 – Console do Cloud Storage



Fonte: Elaborado pelo Autor, (2022)