



UNIVERSIDADE FEDERAL DO MARANHÃO

Programa de Pós-Graduação em Ciência da Computação

Evaldo da Silva Andrade

IndoorDSL: Uma abordagem baseada em MDE para o desenvolvimento de mapas de representação do espaço indoor

**São Luís
2020**

UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

EVALDO DA SILVA ANDRADE

IndoorDSL: Uma abordagem baseada em MDE para o desenvolvimento de mapas de representação do espaço indoor

São Luís

2020

EVALDO DA SILVA ANDRADE

IndoorDSL: Uma abordagem baseada em MDE para o desenvolvimento de mapas de representação do espaço indoor

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação UFMA, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Orientador: Samyr Béliche Vale (Orientador)

Doutor em Ciência da Computação - UFMA

São Luís

2020

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).
Núcleo Integrado de Bibliotecas/UFMA

da Silva Andrade, Evaldo.

IndoorDSL: Uma abordagem baseada em MDE para o desenvolvimento de mapas de representação do espaço indoor / Evaldo da Silva Andrade. - 2020.

131 p.

Orientador(a): Samyr Béliche Vale.

Dissertação (Mestrado) - Programa de Pós-graduação em Ciência da Computação/ccet, Universidade Federal do Maranhão, São Luís, 2020.

1. Domain Specific Language. 2. Indoor. 3. MDE. 4. Modelo de Representação do Espaço Indoor. 5. Navegação Indoor. I. Béliche Vale, Samyr. II. Título.

EVALDO DA SILVA ANDRADE

IndoorDSL: Uma abordagem baseada em MDE para o desenvolvimento de mapas de representação do espaço indoor

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFMA, como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

Aprovado em 24 de setembro de 2020.

BANCA EXAMINADORA

Samyr Béliche Vale (Orientador)

Doutor em Ciência da Computação - UFMA

Carlos de Salles Soares Neto

Doutor em Informática - UFMA

Raimundo Santos Moura

Doutor em Engenharia Elétrica e de Computação - UFPI

*"Aos meus pais, irmãos, amigos, esposa e
filho e aos meus professores. Todos
contribuíram para que esse resultado fosse
alcançado."*

Agradecimentos

Primeiramente agradeço a Deus pela realização pessoal e pela oportunidade de aprendizado e realização profissional.

Agradeço aos meus pais, Maria Aristide da Silva Andrade e Francisco Chagas de Andrade, pelo amor em todas as ocasiões e educação ao longo da vida.

Aos irmãos e demais familiares que sempre torceram pelo meu sucesso.

A minha esposa Regina Costa Nunes Andrade e meu filho Rui Nunes Andrades, por seu apoio incondicional fornecido em todos os momentos, alegrias e motivação ao longo dessa jornada e da vida.

Por todos aqueles que me apoiaram e forneceram inspiração e motivação nos momentos de adversidades, por sempre me dar força para o meu crescimento e assim poder alcançar minhas realizações.

Agradeço a todos os professores do PPGCC/UFMA que me ofereceram seus ensinamentos, conselhos e em especial ao professor Samyr Béliche Vale pela orientação, confiança, paciência e sempre se fazer presente no momento das dúvidas e pela motivação contínua no decorrer do curso.

Aos colegas e amigos do mestrado que compartilharam essa jornada. Em especial aos membros do LSDi com os quais eu pude aprender e compartilhar vários momentos.

Aos meus colegas do Instituto Federal do Tocantins –IFTO campus Araguaína que possibilitaram que eu me mantivesse afastado das funções para que pudesse concluir os créditos relacionados as disciplinas.

E por fim, a Universidade Federal do Maranhão (UFMA) pela infraestrutura e por possibilitar a realização desse trabalho.

*“A tarefa não é tanto ver aquilo que
ninguém viu, mas pensar o que
ninguém ainda pensou sobre aquilo que
todo mundo vê.”*

(Arthur Schopenhauer)

Resumo

Modelos de representação do espaço indoor podem fornecer importantes informações sobre os aspectos arquiteturais e semânticos de um dado prédio. O que viabiliza o uso dessas informações para fomentar o desenvolvimento de aplicações voltadas a navegação e posicionamento dentro desse tipo de ambiente, aplicações que são suportadas por meio de sistemas de posicionamento indoor que surgem visando otimizar a navegação, ou mesmo aumentar a produtividade, ou auxiliar pessoas com deficiência, dentre outras soluções. No entanto, informações acerca do ambiente e ferramentas que possibilitem a elaboração do modelo de representação do ambiente indoor que contenham as informações que possam ser utilizadas para alimentar os sistemas de navegação e posicionamento dentro desses ambientes nem sempre se encontram disponíveis. Ou quando disponíveis se tornam proibitivas em virtude do aparato logístico necessário para a concepção do modelo por meio da utilização de diversos sensores e robôs, ou demandam um esforço significativo através de ferramentas de edição que unem informações geográficas e informações arquiteturais por meio da edição manual e repetitiva de primitivas que compõe esse ambiente, fazendo com que a tarefa seja bastante propensa a erros. Diante disso, este trabalho apresenta a proposta de criação do modelo de representação do espaço indoor através de uma *Domain Specific Language* (DSL), que por meio de uma sintaxe gráfica com aspectos do domínio, fornece a possibilidade de modelagem do ambiente indoor em um nível de abstração elevado com a finalidade de possibilitar que usuários com nenhum ou pouco conhecimento sobre o domínio possam criar seus próprios modelos e realizar transformações entre diferentes modelos. Sendo assim, seu desenvolvimento se dá no escopo da INavigS, uma infraestrutura de software voltada a posicionamento e navegação indoor. Que apesar de prover inúmeras funcionalidades, ainda apresenta algumas lacunas, como no caso da elaboração do modelo de representação do espaço indoor. Ao se aplicar a linguagem no contexto do INavigS, é obtido o feedback dos usuários quanto ao uso da

linguagem, identificando-a como uma ferramenta satisfatória para a criação dos mapas que possam vir a representar o ambiente indoor.

Palavras-chave: Domain Specific Language; Indoor; MDE; Modelo de Representação do Espaço Indoor; Navegação Indoor

Abstract

Indoor space representation models can provide important information about architectural and semantic aspects of a given building. Which enables the use of this information to encourage the development of applications aimed at navigation and positioning within this type of environment, applications that are supported through indoor positioning systems that appear to optimize navigation, or even increase productivity, or aid people with disabilities, among other solutions. However, information about the environment and tools that enable the elaboration of the indoor environment representation model that contain the information that can be used to feed the navigation and positioning systems within those environments are not always available. Either when available they become prohibitive due to the logistical apparatus necessary for the design of the model, or they demand a significant effort through editing tools that combine geographic and architectural information through manual and repetitive editing of primitives that make up this environment, making the task become quite error prone. Therefore, this work presents the proposal of creating indoor space representation through Domain Specific Language, that through a graphical syntax with aspects of the domain, provides the possibility of indoor environment modeling at a high level of abstraction in order to enable users with none or regular knowledge about the domain to create their own models and perform transformations between different models. Therefore its development takes place within the scope of INavigS, a software infrastructure focused on indoor positioning and navigation. Despite providing numerous features, still presents some gaps, as in the case of the elaboration of the indoor space representation model. When applying the language in the context of INavigS, feedback is obtained from users regarding the use of language, identifying it as a satisfactory tool for maps creation which may represent the indoor environment.

Key-words: Domain Specific Language; Indoor; MDE; Indoor Space Representation

Model; Indoor Navigation

Lista de Siglas

ADE *Indoor Application Domain Extensions*

AQL *Acceleo Query Language*

BIM *Building Information Modeling*

BLE *Bluetooth Low Energy*

CAD *Computer-Aided Design*

CIM *Computational Independent Model*

CityGML *City Geography Markup Language*

DAO *Data Access Object*

DSL *Domain Specific Language*

EMF *Eclipse Modeling Framework*

GIS *Geographic Information System*

GMF *Graphical Modeling Framework*

GPS *Global Positioning System*

HTML *HyperText Markup Language*

IFC *Industry Foundation Classes*

IMU *Inertial Measurement Unit*

INavigS *Indoor Navigation System*

IoT *Internet of Things*

IPS *Indoor Positioning System*

ISO *International Organization for Standardization*

JOSM *Java Open Street Map*

JSP *JavaServer Pages*

LAC *Laboratory for Advanced Collaboration*

LBS *Location Based Systems*

LOD *Level of detail*

LSDi *Laboratório de Sistemas Distribuídos Inteligentes*

MDA *Model-Driven Architecture*

MDD *Model-driven Development*

MDE *Model-Driven Engineering*

M-Hub *Mobile Hub*

MOF *Meta-Object Facility*

OGC *Open Geospatial Consortium*

OMG *Object Management Group*

OSM *Open Street Maps*

PIM *Platform Independent Model*

POI *Point of Interest*

PSM *Platform Specific Model*

PSSUQ *Post-Study System Usability Questionnaire*

PUC-Rio Pontifícia Universidade Católica do Rio de Janeiro

RFID *Radio Frequency Identification*

RSSI *Received Signal Strength Indicator*

SLAM *Simultaneous Localization and Mapping*

SQL *Structured Query Language*

SUS *System Usability Score*

UFMA Universidade Federal do Maranhão

UUID *Universally Unique Identifier*

VGI *Volunteered Geographic Information*

WLAN *Wireless Local Area Network*

XMI *XML Metadata Interchange*

XML *Extensible Markup Language*

Lista de Figuras

2.1	Processo de edição do mapa por meio do <i>Java Open Street Map</i> (JOSM). Fonte (LEARNOSM, 2020).	28
2.2	Mapa indoor na forma de tubos com informações em tempo real. (NOS- SUM, 2011).	31
2.3	Relação entre diferentes acrônimos dentro do universo de desenvolvimento baseado em modelos, adaptação de (BRAMBILLA et al., 2017).	35
2.4	Relação Modelo x Metamodelo adaptação de (BRAMBILLA et al., 2012).	36
2.5	Modelos, Metamodelos e Meta-metamodelos (BRAMBILLA et al., 2017).	37
2.6	Amostra de código como exemplo do uso de DSL interna (NEGASH et al., 2017).	40
2.7	Transformações e suas principais utilizações. Adaptação de (COMBE- MALE, 2008).	42
2.8	Exemplo de transformação entre o modelo “Silva family” (<i>source</i>) para o modelo “Person” (<i>target</i>). Adaptação de (JOUAULT, 2013; WAGELAAR, 2018; KHAN, 2010).	43
2.9	Utilizando template para transformação M2T.	48
2.10	Visão geral da arquitetura aplicada a diferentes domínios, adaptação de (MEDEIROS, 2018).	51
2.11	Arquitetura da infraestrutura de software INavigS, adaptação de (MEDEI- ROS, 2018)	51
2.12	Modelo Hierárquico do Espaço Indoor.	54

2.13	Construção do modelo espacial de (MEDEIROS, 2018).	54
4.1	Metamodelo da linguagem de modelagem IndoorDSL.	66
4.2	Partindo de metamodelos até a interface de diagramas do usuário. Adaptação de (KOUHEN, 2013).	68
4.3	Sirius Viewpoint da linguagem IndoorDSL.	69
4.4	Etapas do processo de construção gráfica da linguagem IndoorDSL.	70
4.5	Definição dos elementos gráficos contidos na linguagem IndoorDSL.	70
4.6	Definição visual do elemento <code>Link</code> .	71
4.7	Erro resultante da expressão que avalia a validade do <code>Link</code> .	73
4.8	Erro resultante ao definir um nome já existente a um prédio.	74
4.9	<i>Screenshot</i> da interface de modelagem da linguagem IndoorDSL.	75
4.10	Visão geral do processo de desenvolvimento dos mapas utilizados pelo INavigS.	76
4.11	Fragmento do template destinado a geração para <code>BuildElement</code> e o respectivo código.	77
4.12	Fragmento do template destinado a geração para <code>Beacon</code> e o respectivo código.	77
4.13	Fragmento do template destinado a geração para <code>Link</code> e o respectivo código.	78
5.1	Perfil de escolaridade e atuação dos participantes.	84
5.2	Nível de conhecimento obtido a partir a aplicação do formulário pré-experimento.	84
5.3	Nível de conhecimento sobre modelo de representação do espaço indoor dentre os participantes da área de computação.	85
5.4	Metodologia de aplicação dos experimentos.	86

5.5	Modelo de referência proposto para a primeira tarefa.	90
5.6	Modelo de referência proposto para a segunda tarefa.	93
5.7	Boxplot criado contendo os dados do tempo de duração das tarefas.	96
5.8	Relação entre os tempos obtidos pelos participantes x especialista.	97
5.9	Quantidade de participantes X Tarefas resolvidas em sua totalidade.	98
5.10	Resultado PSSUQ por categorias.	99
5.11	Resultado obtido pelo SUS <i>score</i>	99
A.1	Formulário pré experimento - página 1.	116
A.2	Formulário pré experimento - página 2.	117
A.3	Formulário pré experimento - página 3.	118
A.4	Formulário pré experimento - página 4.	119
A.5	Formulário pré experimento - página 1.	120
A.6	Formulário pré experimento - página 2.	121
A.7	Formulário pré experimento - página 3.	122
A.8	Formulário pré experimento - página 4.	123
A.9	Formulário pré experimento - página 4.	124
A.10	Formulário pré experimento - página 4.	125
B.1	Modelo referente a tarefa fácil.	126
B.2	Modelo referente a tarefa fácil.	126
C.1	Código gerado para o modelo referente a tarefa fácil.	127
C.2	Código gerado para o modelo referênte a tarefa difícil.	128

Lista de Tabelas

1.1	Tabela contendo as fases de desenvolvimento da dissertação.	24
2.1	DSL: Vantagens e Desvantagens (KRUK, 2015)	38
3.1	Tabela comparativa entre os trabalhos correlatos.	60
4.1	Elementos extraídos após a análise da plataforma INavigS.	63
4.2	Elementos utilizado na notação gráfica	72
5.1	Tempo gasto utilizado como referência na realização das tarefas propostas.	82
5.2	Tabela com o <i>checklist</i> de aplicação do experimento.	87
5.3	Tabela com as pontuações referentes ao formulário <i>Post-Study System Usability Questionnaire</i> (PSSUQ).	88
5.4	Dados para Address - Tarefa fácil.	91
5.5	Informações do elemento Build - Tarefa fácil.	91
5.6	Informações relacionadas ao BuildElements - Tarefa fácil.	91
5.7	Informações relacionadas aos <i>Beacons</i> - Tarefa fácil.	91
5.8	Informações relacionadas ao Link.	92
5.9	Dados para Address-Tarefa 2.	94
5.10	Informações do elemento Build	94
5.11	Informações relacionadas ao <i>Beacons</i> - Tarefa 2.	94
5.12	Informações relacionadas ao Link.-Tarefa Difícil	95

5.13	96
5.14 Tempo gasto por participante em relação aos experimentos.	97

Sumário

Lista de Siglas	8
Lista de Figuras	11
Lista de Tabelas	14
1 Introdução	19
1.1 Objetivos	22
1.1.1 Objetivos específicos	23
1.1.2 Metodologia	23
1.2 Estrutura da Dissertação	25
2 Fundamentação Teórica	27
2.1 Mapas para ambiente indoor	27
2.1.1 Representação do espaço	29
2.1.2 Modelos de representação próprios	30
2.2 Engenharia Dirigida por Modelos	32
2.2.1 Arquitetura Dirigida por Modelos	34
2.2.2 Metamodelagem	35
2.3 Linguagem Específica de Domínio	38
2.3.1 DSL interna	39
2.3.2 DSL externa	40

2.4	Transformações entre modelos	41
2.5	Ferramentas de metamodelagem	44
2.5.1	EMF	44
2.5.2	Sirius	46
2.5.3	Acceleo	47
2.6	Indoor Navigation System (INavigS)	49
2.6.1	Arquitetura do INavigS	50
2.6.2	Representação do ambiente indoor utilizada pelo INavigS	53
2.7	Considerações Finais	55
3	Trabalhos Relacionados	56
4	Proposta	61
4.1	Visão geral	61
4.2	Requisitos e decisões de projeto	64
4.2.1	Requisitos do projeto	64
4.3	Sintaxe Abstrata	65
4.4	Sintaxe Concreta	67
4.5	Transformações	75
4.6	Considerações finais	78
5	Avaliação	80
5.1	Experimento	80
5.1.1	Objetivos e Métricas	81
5.1.2	Escolha dos participantes	83

5.1.3	Definição e planejamento da realização dos experimentos	85
5.1.4	Procedimento de análise	87
5.1.5	Ambiente	89
5.2	Tarefas	89
5.2.1	Difícil	92
5.3	Resultados	94
5.4	Ameaças à validade	99
5.5	Considerações Finais	102
6	Conclusão, Publicações e Trabalhos Futuros	103
6.1	Contribuições do trabalho	103
6.2	Limitações da proposta	104
6.3	Publicações	104
6.4	Trabalhos Futuros	105
	Referências Bibliográficas	107
A	Questionários aplicados ao longo da pesquisa	116
A.1	Questionário Pré Experimento	116
A.2	Questionário Pós Experimento	120
B	Modelos desenvolvidos por meio da IndoorDSL referentes as tarefas propostas	126
C	Artefatos resultantes da transformação dos modelos elaborados nas tarefas.	127

1 Introdução

Nas últimas décadas, temos vivenciado um aumento significativo da população mundial que chegou a alcançar a marca de 7.7 bilhões em 2019 (ROSER; ORTIZ-OSPINA, 2013). Paralelamente a esse aumento, podemos observar o aumento da quantidade de prédios e residências existentes nas cidades, já que as pessoas necessitam de um abrigo para que possam permanecer seguras e confortáveis. Além disso, grande parte do tempo gasto cotidianamente por uma pessoa é dentro de algum tipo de edificação, chegando a alcançar patamares entre 80% a 90% do tempo diário (LI; LEE, 2010).

Juntamente ao aumento da quantidade de edificações há a demanda de informações espaciais a respeito desses ambientes nos quais as pessoas estejam inseridas, a fim de prover serviços de localização e orientação espacial para que se possa explorar o ambiente da forma mais inteligente possível. Fazendo com que navegação e determinação de localização de usuários em um ambiente indoor se tornasse um campo atrativo de pesquisas nos últimos anos (SAKPERE et al., 2017).

Os serviços baseados em localização são largamente utilizados hoje em dia, seja pela busca de um endereço, estimativas de distância, ou ao utilizar um aplicativo de transporte de passageiros. Ao interagir com as informações contextuais, provenientes do usuário, esses sistemas podem fornecer a localização e possíveis rotas, para que se possa alcançar um destino a partir de um ponto de origem. Em se tratando de navegação outdoor diversas tecnologias tem sido empregadas para compor o *Location Based Systems* (LBS) e auxiliar na determinação da localização e prover rotas aos usuários, tendo como as principais o *Global Positioning System* (GPS)¹, GALILEO², GLONASS³ e Beidou⁴. No entanto, quando aplicadas na navegação dentro de prédios, não se comportam de maneira

¹<https://www.gps.gov/systems/gps/>

²<https://www.gsc-europa.eu/galileo/what-is-galileo>

³<https://www.glonass-iac.ru/en/GLONASS/documents.php>

⁴<http://en.beidou.gov.cn/SYSTEMS/System/>

satisfatória, devido a imprecisão na estimativa de posição.

Em face aos problemas ocasionados da falta de informações à respeito do ambiente indoor surge o *Indoor Positioning System* (IPS) como um sistema baseado em localização voltado aos ambientes internos das edificações, se utilizando de tecnologias alternativas tais como Bluetooth (SATAN, 2018), *Wireless Local Area Network* (WLAN) (SCHÄFERMEIER et al., 2018) ou *Radio Frequency Identification* (RFID)(HUANG et al., 2015) em vez do uso de um sistema tradicional no qual a localização é determinada a partir da comunicação com um satélite (GPS, GLONASS ou BeiDou), uma vez que em ambientes em que usuário esteja cercado por obstáculos (típico ambiente indoor) pode não funcionar de maneira satisfatória (Ngui; Lee, 2019).

Para que os sistemas de localização e navegação em ambientes indoor possam prover suas funcionalidades é necessário ainda um modelo de representação do espaço indoor que forneça um mapa contendo informações espaciais e topológicas a respeito do ambiente. Em consideração aos modelos de representação de um ambiente *indoor*, há algumas propostas na literatura com o intuito de oferecer uma forma de representação do espaço, armazenamento do modelo e forma de acesso aos dados. Entre elas, se destacam a *Industry Foundation Classes* (IFC)⁵, *CityGML Indoor Application Domain Extensions* (ADE) (KIM et al., 2013) e *IndoorGML*⁶, sendo que os dois últimos são propostos pela *Open Geospatial Consortium* (OGC)⁷. Cada uma delas pode ser utilizada para representar detalhes a nível de arquitetura e geometria, contudo somente o *IndoorGML* fornece suporte a representação dos elementos contidos em um prédio de maneira que possam ser organizados a partir de uma relação topológica. Uma importante característica para aplicações de IPS que tem por objetivo o fornecimento de navegação pelo ambiente.

Em meio a diversas ferramentas que podem ser utilizadas como IPS, surge o *Indoor Navigation System* (INavigS) como uma infraestrutura de software ciente de contexto desenvolvida no Laboratório de Sistemas Distribuídos Inteligentes (LSDi) da Universidade Federal do Maranhão (UFMA) com capacidade de estimar a localização e

⁵www.ifcwiki.org/

⁶<http://www.indoorgml.net/>

⁷<https://www.ogc.org/about>

guiar usuários dentro de um ambiente indoor a partir do uso de um dispositivo móvel (*smartphone*) e as interações que ele realizar com sensores espalhados pelo ambiente físico por meio de conexões bluetooth.

Embora possamos encontrar diferentes modelos e formas de obtenção do mapa de representação do espaço indoor de maneira que possa vir a compreender os elementos arquiteturais bem como os sensores espalhados pelo ambiente, o INavigS possui um modelo próprio que garante que o ambiente possa ser mapeado mesmo em ocasiões em que não se disponha de uma planta arquitetural contendo tais informações para guiar o processo de elaboração do modelo de representação do espaço interno. No entanto foram identificados alguns problemas relacionados a elaboração do mapa contendo todas as informações necessárias para que a ferramenta possa prover os serviços por ela oferecidos, sendo os principais:

- A tarefa de inserção das informações a respeito dos elementos arquiteturais atualmente é restringida ao desenvolvedor da infraestrutura de software, que de forma manual realiza a inserção de todas as informações de coordenadas dos sensores utilizados para estimar a localização (*beacons Bluetooth* instalados pela edificação), bem como as demais informações que constituem o modelo utilizado para representar o espaço indoor em uma base de dados relacional por meio de declarações utilizando a linguagem *Structured Query Language* (SQL).
- A medida com que a quantidade de elementos contidos no prédio cresce, a tarefa de inserção torna-se exaustiva e com uma maior possibilidade de erros, devido a repetição das tarefas manuais.
- Tem o seu uso dificultado por não oferecer uma interface para que usuários sem conhecimentos sobre computação possam inserir suas informações, e assim poderem criar representações dos seus próprios espaços contendo elementos do prédio e sensores, e assim, poder usufruir dos benefícios fornecidos pela infraestrutura de software.

Dessa forma, foi buscado alternativas que possam fornecer suporte ao desenvolvimento do modelo de representação do espaço indoor utilizado pela infraestrutura INavigS. De maneira que possa vir a englobar todas as classes de usuários que porventura venham a fazer uso da infraestrutura para suas aplicações de localização e navegação. Surge a engenharia dirigida por modelos surge como um paradigma de desenvolvimento atrativo, para que se possa reduzir a complexidade associada a criação do mapa contendo o modelo de representação do espaço indoor. Nesse paradigma, o nível de abstração é elevado através do uso de modelos e faz com que usuários possam se preocupar apenas com elementos voltados a modelagem do domínio, sem a necessidade de detalhes da sua implementação tecnológica (BENOUDA et al., 2016), focando somente no modelo que se pretende alcançar.

Tais conceitos permitem desenvolver uma DSL que possibilitará que experts e usuários finais do INavigS possam modelar o ambiente indoor através de uma linguagem específica, dotada de elementos advindos do próprio domínio e que por meio de uma interface amigável, possa se desenvolver os modelos de forma rápida, menos propensa a erros e de fácil manutenção (GARCÍA et al., 2014).

1.1 Objetivos

O objetivo dessa dissertação é a concepção de uma DSL voltada ao desenvolvimento de modelos de representação do espaço indoor utilizado pela infraestrutura de Software INavigS. Fazendo uso de conceitos de metamodelagem, modelos e transformação entre modelos defendidos pelo paradigma de engenharia dirigida por modelos aplicados no âmbito do INavigS.

A representação gráfica visa a redução da dificuldade na inserção de dados que possam vir a representar o modelo espacial do ambiente utilizado pelo INavigS, e assim fornecer ao usuário final, uma ferramenta que torne a elaboração dessa tarefa menos exaustiva, como no caso da concepção do modelo na forma puramente textual.

1.1.1 Objetivos específicos

Para que se possa fornecer uma linguagem de modelagem do ambiente indoor aplicada ao escopo da infraestrutura do INavigs, se faz necessário traçar os seguintes objetivos específicos:

- Identificar os principais conceitos relacionados ao domínio.
- Identificar as principais ferramentas que possibilitem desenvolver a linguagem de modelagem.
- Desenvolver um metamodelo baseado nas informações que infraestrutura de softwares INavigS utiliza como representação do espaço indoor.
- Criar uma linguagem visual que possibilite a elaboração dos modelos de representação do espaço indoor em alto nível, se utilizando apenas de conceitos utilizados no referido domínio.
- Criar regras de validação que possam ser executadas sobre o modelo.
- Definir a transformação de modelo para texto, permitindo que os dados possam ser manipulados pelo INavigS.
- Avaliar a solução proposta, em relação a utilização das infraestrutura do INavigs.

1.1.2 Metodologia

As fases que culminaram nesse trabalho de dissertação de mestrado são descritas na Tabela 1.1.

1. **Definição do Problema:** momento inicial da pesquisa onde se buscou-se identificar o problema e definição do escopo do projeto de dissertação.
2. **Estudos de conceitos, técnicas e ferramentas:** nessa etapa foi realizada o estudo à cerca da engenharia dirigida por modelos, metamodelagem, principais

Tabela 1.1 – Tabela contendo as fases de desenvolvimento da dissertação.

Fases	Atividades
1	Definição do Problema
2	Estudo de conceitos, técnicas e ferramentas
3	Análise do domínio e extração dos elementos do Metamodelo
4	Implementação da linguagem no âmbito do INavigS
5	Modelagem de ambientes indoor
6	Validação através de cenários
7	Conclusões

transformações entre modelos, linguagem específica de domínio. Bem como identificação e escolha das principais ferramentas existentes que pudessem contribuir para a concepção desse trabalho. Identificando o *Eclipse Modeling Framework* (EMF) para auxiliar no processo de metamodelagem, a utilização do *Sirius* para a construção da forma gráfica oferecida pela linguagem e por último o *Acceleo* como sendo a ferramenta responsável pela realização da transformação entre o modelo gráfico e texto.

3. **Análise do domínio e extração dos elementos do Metamodelo:** Etapa na qual foram extraídos os elementos contidos no metamodelo a partir da análise do domínio no qual se pretende desenvolver a linguagem de modelagem. Identificando os principais elementos que compõe o modelo de representação do espaço indoor utilizados no âmbito do INavigS.
4. **Implementação da Linguagem:** nessa etapa, foi utilizado o EMF para concepção do metamodelo Ecore bem como a geração dos artefatos de código oriundos do metamodelo através da geração automatizada propiciada pelo EMF. Tais artefatos são utilizados quando se instância algum objeto na linguagem de modelagem. Após a concepção do metamodelo, se fez uso do Sirius para o desenvolvimento do aspecto visual da linguagem e mapeando suas relações entre os objetos. Por fim, o *Acceleo* foi utilizado como ferramenta de geração de artefatos baseada em template.
5. **Modelagem de ambientes indoor:** fase que compreende a utilização da DSL desenvolvida para modelagem dos modelos de representação do espaço indoor

utilizados pelo INavigS. O uso se caracteriza como uma fase importante por fornecer a oportunidade de se identificar possíveis erros na elaboração do metamodelo.

6. **Validação através de cenários:** após finalizada a construção da DSL, é chegada a vez de ter contato com usuários no processo de modelagem do espaço indoor por intermédio da interface fornecida pela linguagem. A fim de identificar erros e problemas pontuais que ainda possam permanecer remanescentes no projeto. Para isso foram realizados alguns testes com o intuito de aferir o *feedback* dos usuários junto aos objetivos da ferramenta proposta neste trabalho, e verificar a sua eficácia como uma linguagem de modelagem voltada a criação de mapas de representação do ambiente indoor.
7. **Conclusões:** e por fim, são apresentadas as conclusões a respeito do trabalho bem como os resultados obtidos a partir da fase anterior.

1.2 Estrutura da Dissertação

Esse trabalho está organizado em capítulos conforme descritos a seguir:

- No **Capítulo 1** é apresentado a introdução do trabalho, objetivo e metodologia de trabalho e a forma de organização da dissertação.
- **Capítulo 2** apresenta a fundamentação teórica que foi utilizada para desenvolvimento desse trabalho. Sendo dividida em subseções que abordam a construção de modelos de representação do espaço indoor, aspectos relacionados a engenharia dirigida por modelos e é finalizado com a descrição da ferramenta de localização e navegação INavigS.
- **Capítulo 3** aborda alguns dos trabalhos relacionados ao objeto de pesquisa dessa dissertação de mestrado com o intuito de contrastar com a abordagem proposta por esse trabalho.

- **Capítulo 4** apresenta a linguagem de modelagem junto ao metamodelo, definições de design gráfico utilizado na representação do modelo
- **Capítulo 5** apresenta a avaliação e testes realizados com a linguagem.
- **Capítulo 6** é finalizado o trabalho apresentando as conclusões e propostas de trabalhos futuros que poderão agregar melhorias à abordagem proposta.

2 Fundamentação Teórica

Neste capítulo será apresentado alguns dos fundamentos relacionados ao contexto no qual essa dissertação de mestrado está inserida. Serão abordados conceitos relacionados ao mapeamento e representação do ambiente indoor, Engenharia Dirigida por Modelos, linguagem específica de domínio e a infraestrutura de navegação indoor INavigS.

2.1 Mapas para ambiente indoor

Mapas são um importante mecanismo de navegação bastante utilizado ao longo dos séculos na navegação outdoor por fornecer informações que permitem estimar distância e traçar rotas para que pessoas possam alcançar uma determinada localização. Em sua forma mais básica, pode ser representado em um plano 2D no qual os elementos estão indicados por marcações (pontos de referência) espalhadas pelo modelo de representação (mapa), permitindo que um usuário possa determinar sua posição.

Os avanços na navegação têm sido voltados para ambientes indoor nos últimos anos, impulsionado principalmente por diversas iniciativas e indústria que visam prover informações dentro desses ambientes. Alguns dos principais projetos direcionados principalmente ao mapeamento de universidades, estações de trem e outros locais públicos como uma ferramenta para edição OSM (OSMinedit¹), visualização de ambientes mapeados (Openlevelup²), ou podendo ser uma solução mais corporativa como oferecida pela Steerpath³ voltada a indústria e escritórios a fim de melhorar a produtividade. Podendo apresentar soluções para mapear e visualizar, ou somente visualizar dados de ambientes já mapeados.

¹<https://osminedit.pavie.info>

²<https://openlevelup.net>

³<https://steerpath.com/>

Há ainda diversas formas de obtenção do mapa que possa vir a representar o ambiente indoor que podem ser realizadas de um modo completamente automatizado e com soluções mais sofisticadas com a utilização de sensores laser e câmeras (monocular ou binocular). Como proposto por (Chan et al., 2018) que utiliza o algoritmo *Simultaneous Localization and Mapping* (SLAM) com os dados provenientes da fusão de lasers e uma câmera monocular para obtenção do mapa, outras abordagens similares podem ser encontradas em (Liao et al., 2019; Endo et al., 2017). Além disso, o modelo de representação do espaço indoor pode ser obtido a partir do *Open Street Maps* (OSM) utilizando um arquivo *Computer-Aided Design* (CAD), contendo o modelo arquitetural do prédio, e medidas obtidas de forma pessoal e conveniente editadas diretamente em um editor JOSM (ROCHA; ALVES, 2012) ou Potlatch (OSM, 2020), no qual se pode representar toda a geometria existente no ambiente através de nós, caminhos, relações e tags. Conforme ilustrado na Figura 2.1, na qual apresenta o processo de desenho manual dos polígonos que constituem um prédio com a utilização do editor JOSM.

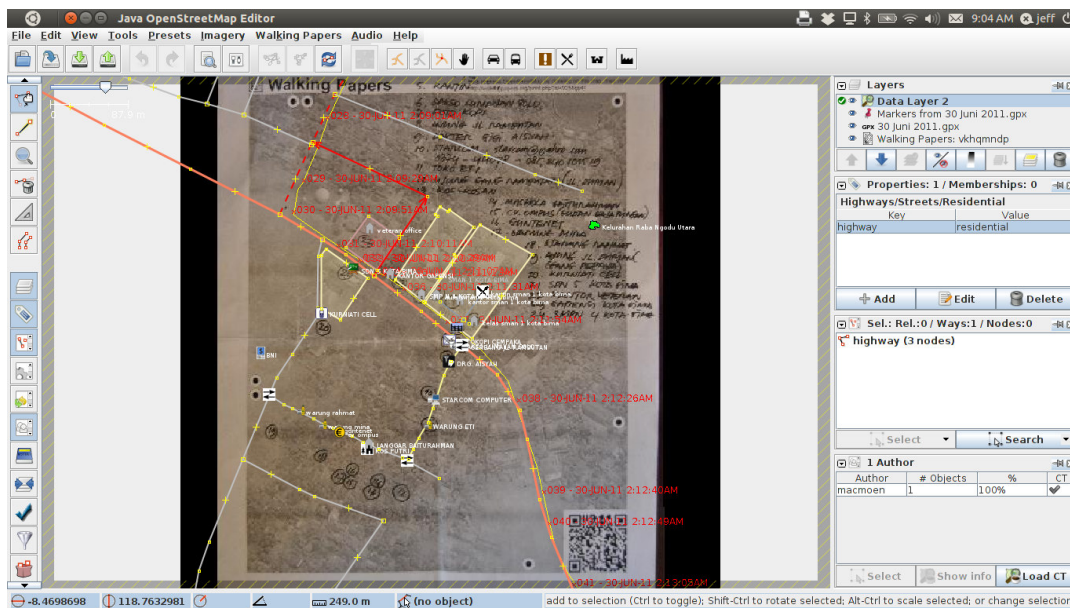


Figura 2.1 – Processo de edição do mapa por meio do JOSM. Fonte (LEARNOSM, 2020).

2.1.1 Representação do espaço

No que diz respeito aos modelos de representação de um ambiente indoor, é possível encontrar diversas propostas na literatura que possuem o intuito de oferecer uma forma de representação, armazenamento e acesso aos dados. Entre elas, se destacam a IFC, *City Geography Markup Language* (CityGML) Indoor ADE e IndoorGML (SRIVASTAVA et al., 2018). Há ainda a possibilidade de se utilizar soluções customizadas para que possam atender a uma necessidade em especial. Uma descrição acerca das principais abordagens para se criar um modelo de representação pode ser vista a seguir, destacando as principais abordagens.

IFC

O Industry Foundation Class IFC são um conjunto de especificações para descrição de informações e troca de dados desenvolvida pela *International Alliance for Interoperability* amplamente aceito no campo da engenharia civil, arquitetura e construção (BRADLEY et al., 2016). Tendo o *Building Information Modeling* (BIM) como principal tecnologia de integração das informações que tem a finalidade de integrar informações de engenharia, construção (materiais utilizados) e operações e manutenção em um único modelo (Lan et al., 2019).

CityGML

O CityGML é uma linguagem proposta pelo OGC destinada a modelagem de cidades dentro do domínio do *Geographic Information System* (GIS), servindo como um padrão universal que apresenta 13 modelos de representação, com 4 níveis de detalhes *Level of detail* (LOD) em cada modelo que são responsáveis por discretizar elementos semânticos e geométricos (CHEN; CLARKE, 2019).

IndoorGML

O indoorGML⁴ é um padrão baseado em *Extensible Markup Language* (XML) proposto pela OGC que tem a finalidade de fornecer um framework comum para representação e troca de informações espaciais presentes no ambiente indoor. No modelo é possível representar informações a respeito de características estáticas e móveis em vez de uma representação focada em apenas elementos arquiteturais. Os conceitos chave descritos em OCG (2019) são:

Espaço das células: o espaço indoor é visto como sendo um conjunto de células que são definidas como a menor unidade de organização e representação do espaço indoor. Contendo algumas propriedades (nome ou outro identificador), limites e coordenadas (nos eixos X, Y, Z) para especificação de sua localização.

Representação Semântica: A semântica é uma importante característica que o IndoorGML utiliza para classificação e identificação da célula e estimar a conexão entre células adjacentes. E assim, definindo quais células serão importantes na navegação (corredores, escadas, elevadores) e quais podem impedir a navegação (paredes e obstáculos).

Representação Geométrica: A representação geométrica, seja em um plano 2D ou 3D não se caracteriza como sendo o principal foco do IndoorGML, por ser definido com mais detalhes em outros padrões com *International Organization for Standardization* (ISO)⁵ ou CityGML.

2.1.2 Modelos de representação próprios

Embora existam diferentes modelos que possam vir a representar um ambiente indoor com diferentes propostas e particularidades, em algumas ocasiões, tais modelos acabam não englobando os aspectos exigidos pelas aplicações, que necessitem de um

⁴<http://www.indoorgml.net/>

⁵<https://www.iso.org>

modelo de representação específico. Isso é devido à ausência de representação dos elementos ou de um determinado recurso que possa vir a ser importante para compor o mapa (e.g. camada de mapa contendo o circuito de câmeras em casos que priorizem a segurança).

A recente busca por um modelo de representação do espaço indoor, tem levado cada vez mais a busca por soluções, que possam mapear o espaço interno de maneira que se ajustem para atender necessidades das aplicações específicas (CHEN; CLARKE, 2019). Esse tipo de soluções que diferem ou complementam os modelos encontrados na literatura são classificados em uma nova categoria de modelos, chamada de formatos de mapas para domínios específicos (Li et al., 2019).

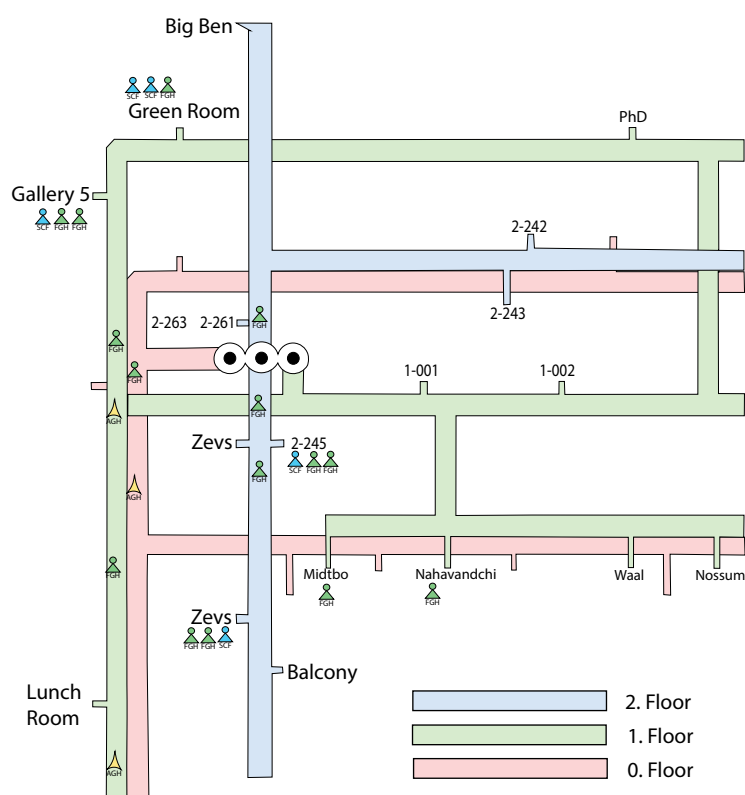


Figura 2.2 – Mapa indoor na forma de tubos com informações em tempo real. (NOSSUM, 2011).

Soluções customizadas podem oferecer vantagens em ocasiões em que os tradicionais modelos não contemplem os objetivos de projetos que porventura se pretende ter, apesar de demandar um maior esforço de desenvolvimento. Uma abordagem específica

pode ser vista no trabalho realizado por Nossum (2011), no qual propôs a criação de mapas indoor destinados a cartografia na forma de tubos, no qual corredores e áreas alcançáveis são modelados na forma de tubulações com cores distintas para representação dos andares. Este modelo, ao ser aplicado a um hospital, permite que o gestor possa identificar e localizar trabalhadores (enfermeiros, médicos e etc.) e pacientes no mapa por diferentes andares e assim facilitar o processo de coordenação dos deveres atribuídos a cada indivíduo dentro do recinto. Um mapa da solução proposta pode ser visto através da Figura 2.2.

2.2 Engenharia Dirigida por Modelos

A utilização de modelos no processo de desenvolvimento de software tem tido amplamente utilizada no ciclo de desenvolvimento de software ao longo das décadas. Sua utilização fornece ao desenvolvedor uma visão abstrata do sistema, na qual o desenvolvedor pode representar diferentes visões à cerca do software, seja a ocorrência de eventos, uma arquitetura ou interação com usuário.

Com o passar dos anos, a atividade de se desenvolver softwares tem se tornado uma tarefa desafiadora em face as inúmeras tecnologias (plataformas e linguagens de programação) disponíveis no mercado. E em meio ao anseio por soluções que pudessem vir a melhorar a produtividade, reduzir os custos e tornar a tarefa de desenvolvimento menos exaustiva para engenheiros de software, surge a *Model-Driven Engineering* (MDE) como um novo paradigma no processo de desenvolvimento de software. Nesse sentido, segundo Cheng et al. (2015), visa reduzir a complexidade associada com o desenvolvimento de software intensivamente complexo, devido ao *gap* entre os conceitos utilizados pelo especialista no domínio, para especificar seus problemas e a abstração de baixo nível fornecida pelas linguagens de programação de propósito geral.

Para Oliveira (2011), a MDE aposta na mudança do foco no processo de desenvolvimento de software para o uso de modelos, em vez de se manter tradicionalmente na atividade de programação, resultando em um aumento da produtividade e um melhor

gerenciamento da complexidade. Sendo que os modelos passam a ter o papel principal no processo de desenvolvimento das aplicações, em vez de ser relegado a uma simples documentação na fase de modelagem. Uma vez que se baseia no desenvolvimento e uso de modelos para gerar artefatos de código, ao separar o conhecimento acerca do domínio do conhecimento sobre programação, através da elevação do nível de abstração, fazendo com que a tarefa de desenvolver softwares se torne mais fácil e atrativa, por não exigir um amplo conhecimento sobre programação, mas apenas o conhecimento relacionado a natureza do domínio no qual se pretende modelar. Para um engenheiro mecânico é mais fácil programar aplicações relacionadas construção de um motor, através de termos que fazem parte do seu domínio (válvulas, bombas de combustível, cabeçote, pistões e etc.) do que se utilizar termos relacionados a uma linguagem de programação.

Na literatura, podemos encontrar diferentes definições para modelos, mas em geral, pode ser compreendido como sendo uma representação simplificada da realidade, obtida através do processo de abstração (MOLINA et al., 2013), que permite se fazer predições e inferências (KÜHNE, 2006). A MDE tendo o modelo como principal ativo, separando o processo de negócio da sua implementação tecnológica, e assim garantindo a interoperabilidade do modelo através da implementação em diferentes escolhas possíveis de tecnologia, que Einarsson et al. (2017) elenca as principais vantagens, como sendo:

- Melhorar a qualidade por aumentar a rastreabilidade entre artefatos, permitindo a detecção de defeitos de forma antecipada, reduzindo erros pela automatização do processo de geração dos artefatos desejados.
- Aumentar a produtividade por meio da maximização da compatibilidade entre sistemas, simplificando o processo de desenvolvimento, e promovendo a comunicação entre equipes de desenvolvimento.
- Facilitar a exploração de modelos a fim de simular, estimar, entender, comunicar e gerar código.

Na próxima subseção será descrita a MDA, uma abordagem baseada em MDE

que fornece um conjunto de padrões, para se descrever e realizar transformações entre modelos.

2.2.1 Arquitetura Dirigida por Modelos

A arquitetura dirigida por modelos (do inglês *Model-Driven Architecture* - MDA⁶) é uma abordagem de desenvolvimento de software, que tem como premissa a utilização sistemática de modelos, através das diferentes etapas do ciclo de vida do processo de desenvolvimento de software. Trata-se de uma arquitetura de metamodelagem desenvolvida em 2001 pela *Object Management Group* (OMG)⁷, sendo uma especialização da MDE que apresenta um guia voltado a estruturação de especificações de softwares, que utilizem modelos ao longo de todo o seu processo de desenvolvimento.

Diversas terminologias para abordagens que tem o modelo como papel principal no processo de desenvolvimento, podem ser encontradas na literatura. Dentre elas, a *Model-Driven Architecture* (MDA), *Model-driven Development* (MDD) e MDE, que embora apresentem nomenclaturas diferentes, elas se inter-relacionam e fazem parte de um universo que faz uso de modelos para tornar a criação de aplicações mais abstrata. A relação entre tais abordagens pode ser representada através do diagrama de Venn exposto na Figura 2.3, no qual $MDA \subset MDD \subset MDE$. Temos a MDE como sendo um conjunto de técnicas e processos de desenvolvimento baseado em modelos. A MDD como um subconjunto da MDE, representando um paradigma de desenvolvimento que tem modelos como artefatos primários e por último a MDA com sendo uma visão particularizada e padronizada da MDD, considerada uma das iniciativas mais proeminentes com aplicação dos padrões regidos pela OMG (HAMMOUDI et al., 2019).

Ao fazer a separação entre a lógica do sistema e seus aspectos de implementação, para uma determinada plataforma a MDA, permite especificar três tipos de modelos distintos sendo (JÁCOME et al., 2018):

⁶<https://www.omg.org/mda/>

⁷www.omg.org

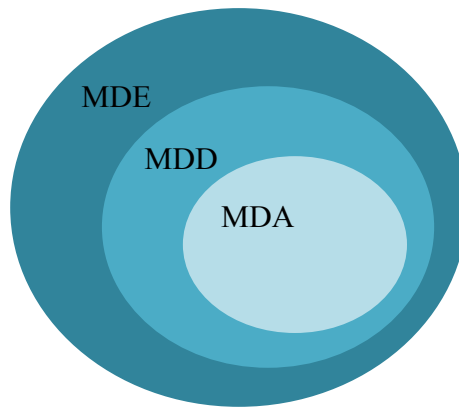


Figura 2.3 – Relação entre diferentes acrônimos dentro do universo de desenvolvimento baseado em modelos, adaptação de (BRAMBILLA et al., 2017).

Computational Independent Model (CIM): sendo o modelo utilizado para descrever a lógica do sistema de forma mais abstrata ao descrever suas funcionalidades sem fazer menção aos aspectos de implementação.

Plataform Independent Model (PIM): Após especificado o CIM, esse modelo vem suprir a necessidade da apresentação de uma perspectiva, na qual se permite expressar a lógica do modelo de negócio, de forma independente da plataforma alvo de sua implementação.

Plataform Specific Model (PSM): nesse modelo é possível referenciar a tecnologia que será utilizada na implementação e implantação do sistema por meio dos aspectos incorporados dos modelos anteriores a ele. Para que sejam gerados todos os códigos oriundos das transformações para a plataforma alvo, sendo importante ressaltar que se pode gerar vários PSMs a partir de um determinado PIM em ocasiões em que há mais de uma plataforma como alvo de implementação. Após especificado permite expressar a lógica independentemente da plataforma alvo.

2.2.2 Metamodelagem

A partir da definição do modelo como sendo o principal agente na MDE, o passo seguinte é definir e representar os próprios modelos como instâncias de algum metamodelo

mais abstrato. Onde metamodelo pode ser definido como modelos de modelos, com a proposta de analisar, construir e definir o conjunto de regras e restrições com aplicação de teorias voltadas a construção de sistema direcionados a determinado domínio (BORELLI, 2016). De modo prático, metamodelos constituem a definição de uma linguagem de modelagem na qual a relação entre modelo e metamodelo pode ser compreendida como uma relação de instanciação, metamodelo descreve modelo, e modelo é uma instância de seu metamodelo. Conforme ilustrado na Figura 2.4

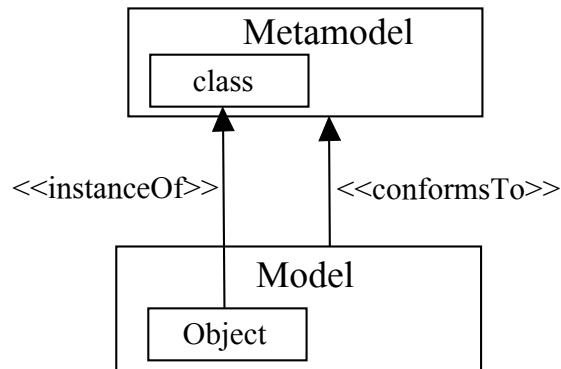


Figura 2.4 – Relação Modelo x Metamodelo adaptação de (BRAMBILLA et al., 2012).

Portanto, usamos um modelo para representar a realidade, e modelos que descrevem modelos (os chamados metamodelos) e recursivamente modelos que descrevem metamodelos (chamados meta-metamodelos). Enquanto na teoria, podemos dispor de uma infinidade de modelos, na prática o modelo proposto pela OMG divide as linguagens de modelagem em uma arquitetura de quatro camadas, que é suportada através da *Meta-Object Facility* (MOF)⁸. O modelo em quatro camadas descrito pela acOMG, pode ser visto na Figura 2.5 com um exemplo que ilustra como a metamodelagem funciona, e a partir de uma abordagem *topdown* se constituindo das seguintes camadas:

M3: trata-se da camada localizada no topo da hierarquia com representação a nível de meta objetos (MOF) chamada de meta-metamodelo, também conhecida como modelo do modelo do modelo e sendo responsável por fornecer definições e conceitos para que se possa especificar uma linguagem de modelagem.

⁸<https://www.omg.org/spec/MOF/2.4.1/About-MOF/>

M2: localizada no segundo nível e conhecida como metamodelo, nessa camada um metamodelo é definido como uma instância do meta-metamodelo a fim de fornecer elementos para que se possa definir uma linguagem de modelagem.

M1: camada modelo onde usuários podem definir seus próprios modelos, em acordo com a sua necessidade para um dado domínio.

M0: define uma instância real e em execução no sistema dos elementos modelados a partir do modelo.

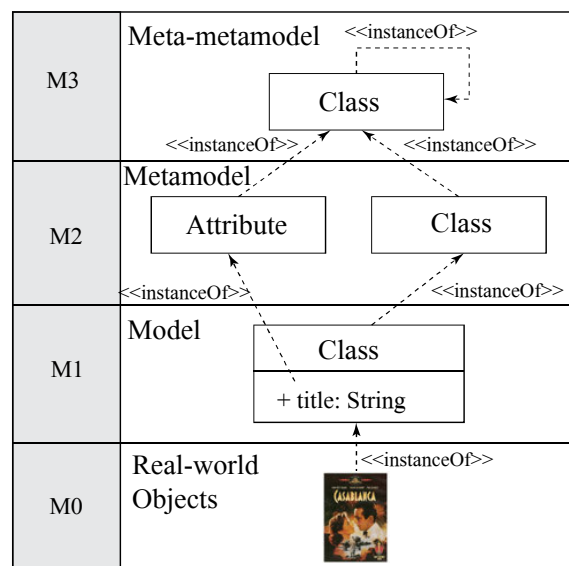


Figura 2.5 – Modelos, Metamodelos e Meta-metamodelos (BRAMBILLA et al., 2017).

A compreensão da hierarquia do modelo em camadas pode ser alcançada de uma melhor forma a partir do entendimento de que ao longo dos níveis, sempre que ocorre a instanciação de um modelo, ela deve ser conforme o seu metamodelo especificado na camada superior. Ou seja, instância de objeto (*instanceOf*) no nível de modelo M1 deve estar em acordo (*ConformsTo*) com o metamodelo M2 descrito na camada acima. O mesmo conceito pode ser aplicado as demais camadas ao longo da arquitetura.

2.3 Linguagem Específica de Domínio

Uma linguagem específica de domínio (do termo em inglês *Domain Specific Language* – DSL) é compreendida como sendo uma linguagem de programação de computadores, com baixa expressividade voltada ao desenvolvimento de aplicações para um domínio em particular (FOWLER, 2010). No entendimento de Deursen et al. (2000), uma DSL é uma linguagem de programação ou linguagem de especificação executável que oferece, através de notações apropriadas e abstrações, expressivamente focada e usualmente restrita a um problema de domínio particular. Na visão de Portugal et al. (2016), uma DSL é uma linguagem desenvolvida para resolver problemas em um domínio específico. Sendo que sua expressividade é o foco na classe de aplicação, que move a engenharia de software para longe de detalhes de baixo nível, por não serem o foco da aplicação. Podemos encontrar diversas definições na literatura para DSL, sendo que em geral tais linguagens apresentam uma sintaxe reduzida, com aspectos que foram captados do domínio no qual se pretende desenvolver aplicações e tornando a tarefa de desenvolvimento de software mais ágil, menos propensa a erros sem que o usuário precise se preocupar com aspectos ligados a uma linguagem de programação específica.

Em (KRUK, 2015), pode-se encontrar uma lista contendo as principais vantagens e desvantagens em desenvolver e usar uma DSL, para que se possa criar aplicações a partir dos modelos especificados contendo aspectos relevantes ao domínio de interesse, conforme listadas na Tabela 2.1.

Tabela 2.1 – DSL: Vantagens e Desvantagens (KRUK, 2015)

Vantagens	Desvantagens
V1- Aumento da Produtividade	D1 - Custo de educação para usuários da DSL
V2 - Programadores e especialistas no domínio	D2 - Complexidade de projetar linguagem e implementação
V3 - Melhora no processo de manutenção	D3 - Manutenção mais difícil
V4 - Aplicações mais leves	D4 - Aplicações mais lentas em caso de compiladores e tradutores complexos.
V5 - Aplicações confiáveis	

Para que se possa definir uma linguagem de modelagem, alguns aspectos devem ser levados em consideração no momento de sua concepção. Para Brambilla et al. (2012), ao se desenvolver uma linguagem há três principais conceitos básicos que são classificados

como sendo:

Linguagem Abstrata: que consiste em uma representação em mais alto nível abstração, sendo responsável por descrever a forma com que as diferentes primitivas que compõem a linguagem podem se inter-relacionar independentemente de qualquer codificação. Se constitui do metamodelo contendo todas os conceitos e abstrações, bem como suas relações.

Linguagem Concreta: após a construção do metamodelo, se pode criar modelos (instâncias) baseados nas especificações descritas na linguagem abstrata. A linguagem concreta é tida como a notação da linguagem, que descreve as representações contidas na linguagem de modelagem permitindo expressar conceituações por meio de notação gráfica ou textual. No caso do uso de linguagens visuais, o produto pode ser um ou mais diagramas contendo o modelo criado.

Semântica: A formalização da sintaxe contida no metamodelo e nas notações não é suficiente para que se possa definir uma linguagem, havendo a necessidade de dar um significado aos elementos constituintes da linguagem. A fim de facilitar a interpretação de suas notações e a correspondência entre os elementos modelados e sua representação abstrata.

As linguagens específicas de domínio podem ser categorizadas como gráfica quando se faz o uso de símbolos extraídos do próprio domínio, para que se possa criar seus modelos, ou textual, ao utilizar uma gramática para determinar a forma com que deve ser redigida pelo usuário na criação dos modelos. As linguagens textuais podem ainda ser classificadas em dois tipos distintos, que segundo Portugal et al. (2016), uma DSL pode ser interna ou externa.

2.3.1 DSL interna

A DSL interna consiste na utilização de um subconjunto de aspectos de uma linguagem mais abrangente, como uma linguagem de propósito geral, mas sendo aplicadas

de forma restrita a um domínio. Essa fração com sintaxe reduzida, representa um código válido na linguagem que à originou. Exemplos desse tipo de DSL podem ser encontradas em (SNEPS-SNEPPE; NAMIoT, 2015), que faz o uso de uma DSL para obter informações de dispositivos de *Internet of Things* (IoT) dentro de aplicações Web tendo como linguagem base o uso de *tags* customizadas e módulos *JavaServer Pages* (JSP). Já o trabalho de (NEGASH et al., 2017) utiliza JavaScript e *HyperText Markup Language* (HTML) para criar uma DSL que forneça modificabilidade e programabilidade em um ambiente de IoT, ao representar os recursos através de uma linguagem de script leve. Uma amostra do código utilizado na linguagem (DSL interna) pode ser observada na Figura 2.6, no qual apresenta fragmentos de duas versões da linguagem proposta por (NEGASH et al., 2017) que tem JSP como linguagem hospedeira, ambos tem a função de fazer com que um determinado dispositivo aguarde pela inicialização de um serviço em um loop contínuo com checagem a cada 12 milissegundos.

```

1
2 !Version .1!
3 while ( device ->ready());
4   device ->every(12){
5     snumber _sid:=4; snumber _aid:=2; snumber _eid:=16;
6     sformat _format := {id, timestamp, unit, value, type };
7     sval _measure := sensor ->read(sid);
8     if (measure->Value >= 12.35){
9       actuator ->rotate(aid,90);
10    } else {
11      while ( event ->getState(eid));
12      if( event ->args(eid) < 0) {
13        device ->accumulate(measure, format);
14      } else {
15        device ->sleep();
16      }
17    }
18  }

```

```

1
2 !Version 0.1!
3 while ( device ->ready());
4   device ->every(0){
5     snumber _eid:=3;
6     snumber _aid:=1;
7     snumber _new:=0;
8     snumber _value:=127;
9     while ( event ->getState(eid));
10    if ( event ->args(eid)>=4) {
11      actuator ->switch(aid,new);
12    } else {
13      actuator ->increase(aid,value);
14    }
15  }

```

Figura 2.6 – Amostra de código como exemplo do uso de DSL interna (NEGASH et al., 2017).

2.3.2 DSL externa

Já uma externa, não é baseada em qualquer linguagem existente para que se possa utilizar como hospedeira, os desenvolvedores podem ter a liberdade de especificar sua sintaxe/gramática, bem como a plataforma alvo da linguagem.

2.4 Transformações entre modelos

Uma das principais vantagens da utilização de modelos no processo de construção das aplicações é a possibilidade de poder transformar um modelo em outro. Um modelo A (*source*) pode ser transformado em um modelo B (*target*), por meio do mapeamento de elementos correspondentes entre os metamodelos, que nortearam seu desenvolvimento. A partir do processo de transformação se pode fazer o reuso dos artefatos a fim de aumentar a produtividade no processo de desenvolvimento das aplicações. Para JÚNIOR et al. (2018), o processo de transformação pode ser realizado por meio de uma das possíveis formas:

Manual: quando o desenvolvedor realiza a tarefa;

Semi-automático: quando o desenvolvedor escolhe quais elementos do modelo devem ser transformados.

Automática: quando não há necessidade da intervenção humana.

Além de ser considerada uma parte essencial no desenvolvimento orientado ao uso de modelos, a transformação entre modelos também é responsável pela automatização de série de operações a serem executadas sobre os modelos nas quais Lano e Rahimi (2013) destaca como sendo:

- Mapear um modelo em um nível de abstração para outro modelo em um nível diferente por meio de refinamento, abstrações ou engenharia reversa.
- Realizar comparações entre dois modelos distintos a fim de identificar diferenças e consistências existentes.
- Gerar fragmentos de código ou outros formatos de texto.
- Propagar mudanças através de modelos que estejam ligados.
- Migrar modelos através das versões existentes com o intuito de permitir que sejam modelados em uma linguagem diferente da utilizada na versão anterior.

As transformações podem ser realizadas de duas formas distintas, podendo ser entre modelos *Model to Model* (M2M) quando se pretende converter um modelo em um outro distinto ou *Model-to-Text* (M2T) quando se pretende converter o modelo em um documento de texto, como por exemplo um código java, HTML, SQL e etc, mesmo que o documento de texto possa ser compreendido como sendo um modelo (STAHL; VÖLTER, 2006). Além disso, elas podem ocorrer de forma horizontal (*transformações horizontais*) como no caso da transformação de um PIM para PSM ou vertical (*transformações verticais*) no caso de um PIM para outro PIM ou de um PSM para outro PSM. Na Figura 2.7 é possível observar as diferentes classes de transformações, bem como a indicação de suas utilizações (COMBEMALE, 2008).

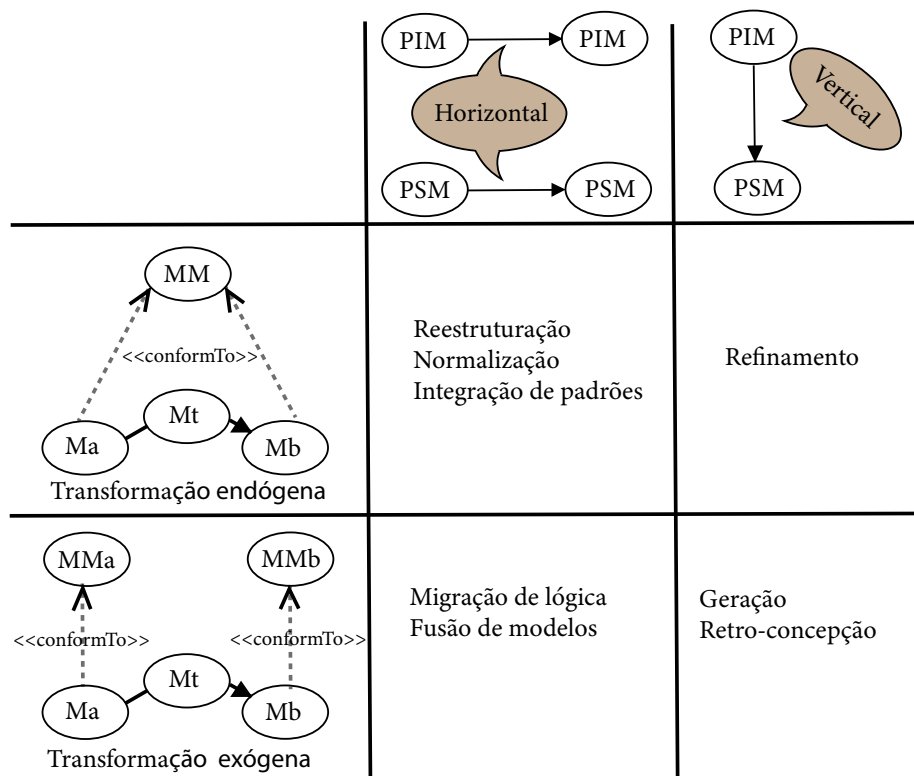


Figura 2.7 – Transformações e suas principais utilizações. Adaptação de (COMBEMALE, 2008).

Conforme abordado anteriormente, o processo de transformação ocorre por meio do mapeamento de correspondências entre dois modelos (*source* e *target*). Tal mapeamento

pode ser realizado por meio da linguagem *Atlas Transformation Language* (ATL)⁹. O ATL é uma linguagem de transformação de modelos que permite definir um modelo de entrada para um modelo de saída através de declarações imperativas ou declarativas. Um exemplo do processo de transformação pode ser visto na Figura 2.8, no qual apresenta no nível M0 o modelo “*Silva Family*” (*source*) em conformidade com o “*FamilyMetamodel*” (lado esquerdo nível M1) e se pretende transformar no modelo “*Person*” (*target*) em mesmo nível, que está em conformidade com o “*PersonMetamodel*” (lado direito nível M1). Ambos os metamodelos estão em conformidade com o meta-metamodelo “*Ecore*” (nível M2), a conversão é possível devido ao conjunto de regras escritas em ATL (expressas no modelo “*Family2Person.atl*”) que mapeia as correspondências entre os metamodelos fonte e alvo.

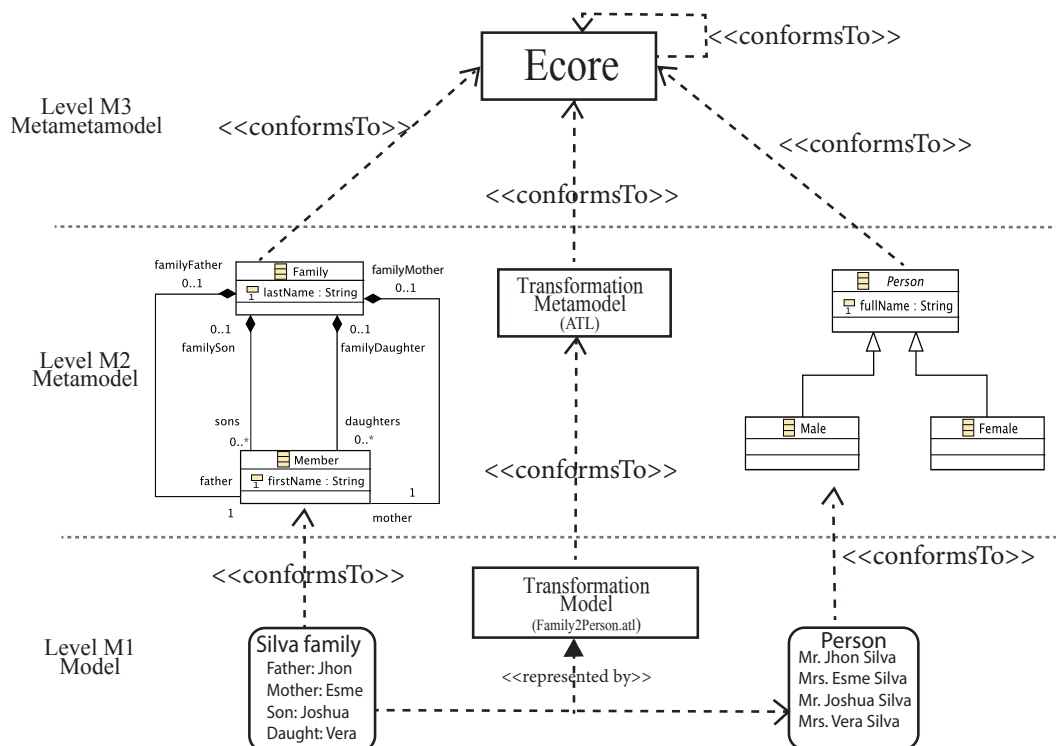


Figura 2.8 – Exemplo de transformação entre o modelo “*Silva family*” (*source*) para o modelo “*Person*” (*target*). Adaptação de (JOUAULT, 2013; WAGELAAR, 2018; KHAN, 2010).

⁹<https://www.eclipse.org/atl/>

2.5 Ferramentas de metamodelagem

Nesta seção é apresentado algumas das ferramentas que podem ser utilizadas para o desenvolvimento de uma linguagem, seguindo os princípios da metamodelagem.

2.5.1 EMF

O EMF é um framework integrado ao ambiente de desenvolvimento eclipse, que gera ferramentas baseadas no modelos estruturados para geração de código (ROSA et al., 2013). Os metamodelos podem ser criados através do uso das ferramentas EMF `Ecore` como linguagem de metamodelagem, mesma extensão dos arquivos gerados que contém a representação do metamodelo.

Em um típico metamodelo `Ecore`, há quatro construtores básicos que são necessários para que se possa representar um modelo através de uma linguagem de modelagem. Que segundo Gasevic et al. (2009) são compostos basicamente por:

EClass: representação da meta-classe utilizada na modelagem de classes, possuindo nome e podendo fazer referências a 0 ou mais atributos, e também a 0 ou mais referências.

EAttribute: utilizado para representar um atributo modelado, sendo constituído de nome e tipo de dado. Podendo referenciar diferentes tipos de dado (string, double, float, int, boolean e etc).

EReference: utilizada para representar as diferentes relações existentes entre classes (herança, composição, associação por exemplo).

EDataType: utilizados para representar tipos de atributos, podendo ser um tipo primitivo (int, float) ou do tipo objeto (java.Util.Date).

Por ser uma linguagem de metamodelagem o `Ecore` permite elevar o nível de abstração para a especificação de uma DSL por meio dos metamodelos. Além do fato de

se poder gerar artefatos de código escritos em Java a partir da tradução dos metamodelos de forma direta e automatizada através da transformação M2T. Para Borelli (2016) o núcleo do EMF é composto por três principais partes como sendo:

EMF: consistindo do framework que contém o metamodelo na extensão `Ecore` utilizado essencialmente para elaboração de metamodelos.

EMF.edit: framework contém as classes reutilizáveis que viabilizam o desenvolvimento de editores que manipulam modelos gerados a partir do EMF.

EMF.codegen: gerador de código capaz de gerar todo o código necessário para se construir um editor de modelos EMF.

A construção dos metamodelos é realizada através de uma interface gráfica oferecida pelo próprio EMF, onde se permite criar instâncias das meta-classes, atributos e relações que constituem o metamodelo objetivado. Para que seja garantida a interoperabilidade do metamodelo, todas as informações são representadas através de um arquivo *XML Metadata Interchange* (XMI)¹⁰ contendo todos os seus componentes e relações. Após finalizada a construção, o usuário pode gerar automaticamente todo o código relacionado ao metamodelo criado.

O EMF conta ainda com uma série de plugins que ao serem adicionados a ferramenta permitem realizar novas atividades acrescidas das nativas fornecidas pela própria ferramenta. Dentre os principais plugins se destacam *Acceleo*¹¹ (destinado a transformações M2T), *Sirius*¹² (destinado a transformações M2M), *XText*¹³ (utilizado no desenvolvimento de DSL textual).

¹⁰<https://www.omg.org/spec/XMI/>

¹¹<https://www.eclipse.org/acceleo/>

¹²<https://www.eclipse.org/sirius/>

¹³<https://www.eclipse.org/Xtext/documentation/>

2.5.2 Sirius

O Sirius é um plugin que pode ser acrescentado ao EMF de forma gratuita e tem a finalidade de permitir se criar facilmente seu próprio *workbench* para uma linguagem de modelagem gráfica, mediante o aproveitamento das tecnologias EMF e *Graphical Modeling Framework* (GMF). Sendo o GMF¹⁴ responsável por fornecer ferramentas para definição e mapeamento de modelos gráficos.

Ao se utilizar de uma a ferramenta para modelagem gráfica é possível configurar uma visualização personalizada com base no enfoque de pontos de visão (*viewpoints*), que permite personalizar aspectos que são relevantes para usuários de um domínio de negócio, nos casos da modelagem de uma DSL. A especificação de uma representação dedicada e suas ferramentas associadas são feitas pela criação e configuração de um “*Viewpoint Specification Model (VSM)*” (VIYOVIC et al., 2014). Nessa estrutura contém as descrições da estrutura, aparência e comportamento que devem ser obedecidos pelos usuários que irão utilizar para criar seus modelos. Sendo que os principais conceitos associadas a criação de um “*Viewpoint Specification Model*” são (ECLIPSEORG, 2020b):

Viewpoint é o elemento central que o responsável pela especificação do ponto de visão pode definir. Tratando-se de um conjunto lógico de especificações de representação e especificações de extensão de representações.

Representation conjunto de construtores gráficos responsáveis por representar dados do domínio. Sendo que na representação são descritos sua aparência, estrutura e comportamento que o modelo deve seguir. Quanto aos dialetos, eles podem se apresentar na forma de diagramas, tabelas, matrizes e árvores.

Mapping ação responsável por identificar o subconjunto de elementos semânticos pertencentes ao modelo que devem aparecer na representação e como eles devem ser representados. .

Style determina a aparência visual dos elementos. Sendo que um mapeamento pode

¹⁴<https://www.eclipse.org/gmf-tooling/>

possuir diversos estilos atribuídos a ele.

Tool responsável por descrever o comportamento do mapeamento, sem ela o modelo será apenas visualizado sem permitir qualquer processo de edição.

Para que seja configurado as definições dos elementos existentes no modelo, ligações entre elementos e ferramentas disponíveis no processo de modelagem de um modelo é necessário utilizar expressões escritas na linguagem *Acceleo Query Language (AQL)*¹⁵ que também pode ser utilizada para navegação e pesquisa em um modelo Ecore. Além disso, Sirius também pode utilizar outros interpretadores de expressão tais como (ARRUDA, 2019):

Var: utilizada para fornecer acesso direto ao valor de uma variável a partir do seu nome de identificação.

Feature: oferece o acesso a uma determinada *feature* a partir de um elemento selecionado.

Service: utilizado para invocar outros métodos Java que deverão ser executados sobre o atual elemento. Principalmente, em caso em que a ferramenta não atenda às necessidades particulares exigidas para o correto funcionamento da linguagem.

2.5.3 Acceleo

O Acceleo é uma linguagem de transformação utilizada para converter modelos desenvolvidos por meio de uma linguagem de modelagem em texto (transformações do tipo M2T) por meio de templates, visando oferecer algumas vantagens, como a fácil iniciação e familiaridade com a ferramenta, interoperabilidade e a possibilidade da criação de geradores de código customizados (ECLIPSEORG, 2020c).

O princípio é a realização do uso de módulos, que são plugins para geração de código do Acceleo, sendo que cada módulo é um componente de uma tecnologia específica que fornecem soluções com uma alta confiabilidade e de rápida implementação (KARA,

¹⁵<https://www.eclipse.org/acceleo/documentation/>

2015). Um módulo é um arquivo com a extensão “*.mtl” que descreve os dados requeridos para a geração do código fonte na forma de um template, tendo um metamodelo definido para a sua manipulação. O princípio de funcionamento da transformação pode ser observado através da Figura 2.9, onde demonstra as etapas do processo de transformação. (1) o template é lido pelo “*Template Engine*” para que se possa acessar os elementos contidos no modelo (diagrama conforme especificado em um metamodelo), (2) os dados são acessados por meio dos componentes dinâmicos presentes no template através das expressões [aEClass.name/] (para obter o nome da classe). Além disso, templates também oferecem suporte a utilização de estruturas de controle conforme explicitado com o uso do laço [for] que faz a interação sobre todos os atributos contidos na classe através da expressão [for (att : EAttribute | eAllAttributes)] e para cada atributo existente é gerado uma variável resultante na classe Java. (3) a combinação das entradas é processada pelo gerador de código. (4) e por fim é gerado a classe contendo os elementos descritos no modelo.

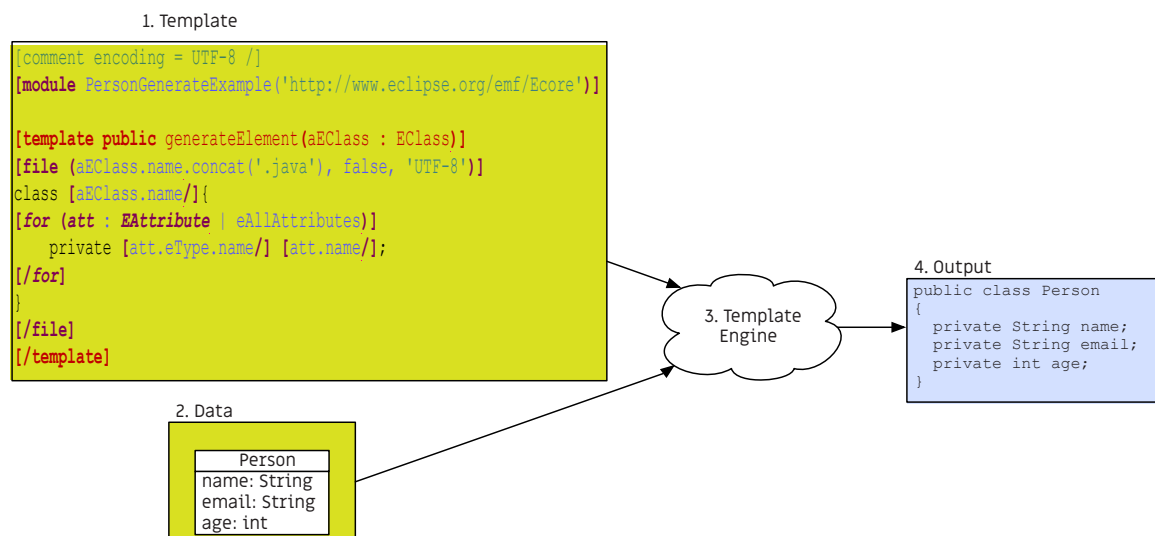


Figura 2.9 – Utilizando template para transformação M2T.

2.6 Indoor Navigation System (INavigS)

O INavigS (MEDEIROS, 2018) é uma infraestrutura de software suportada por dispositivos móveis e com sensibilidade ao contexto, que tem por objetivo o fornecimento de serviços de navegação em um ambiente indoor desenvolvida no LSDi da UFMA. A infraestrutura de software faz uso do middleware de gerenciamento de dados de contexto *Mobile Hub* (M-Hub) (MUNIZ, 2017), desenvolvido em uma parceria entre o *Laboratory for Advanced Collaboration* (LAC)¹⁶ da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio) com o LSDi¹⁷. Sendo que os principais serviços oferecidos pela infraestrutura são:

- Geocodificação: realizando a conversão entre localização geográfica (latitude e longitude) em espaços internos (sala, corredor e etc.).
- Cálculo de rota: obtendo o posicionamento a partir da posição atual do dispositivo móvel, e assim, fornecer orientação ao caminho que deve ser percorrido.
- Provimento de informações sobre o espaço indoor: existência de obstáculos, rampas, elevadores, escadas e etc.
- Assistência durante a rota: indicando as orientações que devem ser seguidas, para que se possa alcançar um destino (dobre a esquerda, dobre à direita, siga em frente) durante todo o percurso.

Para prover suas funcionalidades, o INavigS se aproveita de dispositivos que utilizam a tecnologia *Bluetooth Low Energy* (BLE) dispersados de forma estática pelo ambiente, além do dispositivo móvel (*smartphone*) do próprio usuário (SOUSA, 2019). Uma vez que os beacons estejam posicionados, mapa de representação do ambiente indoor elaborado e o usuário dotado de um *smartphone*, se pode fazer o uso da infraestrutura de software para solicitar instruções para que possa se alcançar um destino a partir

¹⁶<http://www.lac.inf.puc-rio.br/>

¹⁷<http://www.lsdj.ufma.br/>

de um ponto de origem dentro do ambiente indoor. Ao se realizar essa solicitação, o usuário recebe as instruções de navegação para seu deslocamento ao longo do percurso, e à medida que se desloca, uma nova rota é calculada com base na posição atual do dispositivo móvel. Seu funcionamento depende intrinsecamente de conexão com a nuvem por todo o trajeto, uma vez que o processamento dos serviços ocorre externamente ao dispositivo móvel.

Em se tratando de recurso de hardware, o INavigS é composto em sua grande maioria de beacons BLE operados por bateria e posicionados de forma estática no ambiente físico. Esses beacons periodicamente enviam pacotes chamados *advertising packets* contendo sua identificação e o valor da potência do sinal recebido em decibéis. O usuário com o uso de um *smartphone* ao receber esses sinais, pode estimar sua posição e navegar pelo ambiente se baseando nos sinais recebidos dos beacons espalhados ao longo do caminho por ele percorrido. Essa é uma das técnicas de localização baseada em proximidade, na qual se estima a posição a partir do *Received Signal Strength Indicator* (RSSI), no caso do INavigS, o sinal dos beacons *Bluetooth*.

Os recursos de navegação fornecidos pela infraestrutura de software podem ser aplicados nos mais diversificados domínios, como por exemplo, na navegação dentro de um hospital (CHALHOUB; SIMON-NAGY, 2018), em um sistema de localização de mercadorias ou serviços dentro de um estabelecimento comercial (KHALIFA et al., 2010), ou até mesmo guiar turistas pelos principais pontos de uma cidade ou museu (RUBINO et al., 2013). Tais aplicações são evidenciadas pela Figura 2.10, no qual o usuário recebe todas as informações de navegação por meio de um dispositivo móvel.

2.6.1 Arquitetura do INavigS

O INavigS (MEDEIROS, 2018) tem sua arquitetura composta basicamente por duas camadas e quatro componentes distribuídos entre elas. Sendo que cada camada e componente possui um papel intrínseco, uma visão acerca da organização de sua arquitetura e elementos que a compõem pode ser vista na Figura 2.11.

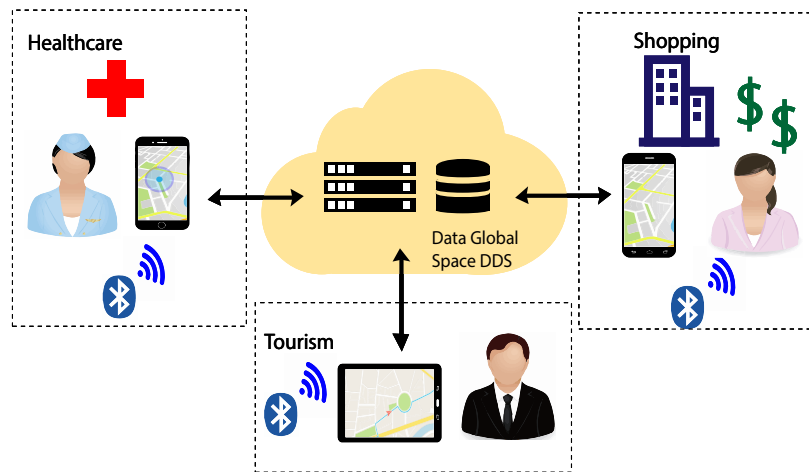


Figura 2.10 – Visão geral da arquitetura aplicada a diferentes domínios, adaptação de (MEDEIROS, 2018).

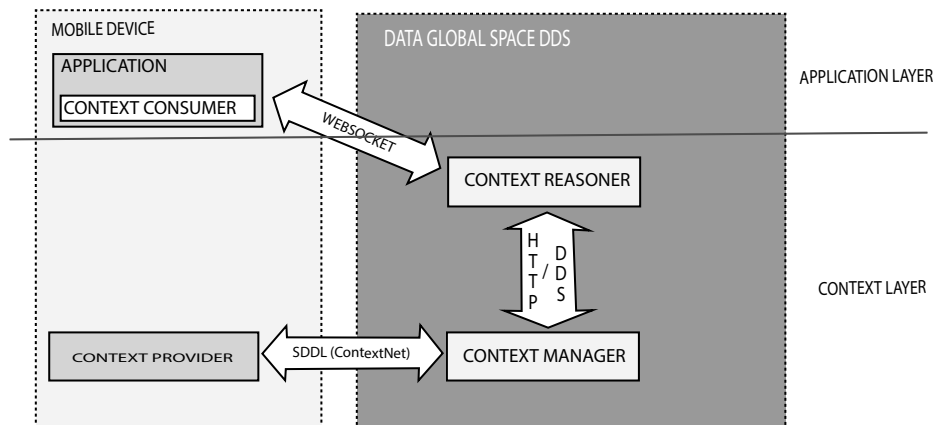


Figura 2.11 – Arquitetura da infraestrutura de software INavigS, adaptação de (MEDEIROS, 2018)

A camada *Context Layer* tem a função de coletar, gerenciar e processar os dados de contexto, nela se encontra os componentes *Context Provider*, *Context Manager* e *Context Reasoner*. Enquanto que na *Application Layer* se encontra as aplicações consumidoras dos serviços providos pela infraestrutura de software, podendo vir a fomentar o desenvolvimento de novas aplicações consumidoras dos serviços de informações de contexto, ela é constituída pelo componente *Context Consumer*. Os quatro componentes presentes na arquitetura possuem os seguintes papéis:

Context Provider: elemento constituinte da *Context Layer*, tal elemento tem a função

de captar, representar e realizar a parte de pré processamento dos dados de contexto. Nesse caso, determinar a localização do usuário no espaço indoor a partir do encontro entre o beacon bluetooth e o *smartphone* do usuário

Context Manager: tem a função de processar e armazenar as informações contextuais, registrando todos os encontros que ocorrem entre beacons e dispositivos móveis. Faz parte do conjunto de componentes que constituem a *Context Layer*.

Context Reasoner: sendo o último elemento que faz parte da *Context Layer*, esse elemento tem a finalidade de prover informações inferidas, a partir dos dados provenientes do contexto. Também é responsável pela conversão de beacons em espaços semânticos, fornecimento de rota, e instruções ao longo do percurso.

Context Consumer: é um mecanismo localizado na *Application Layer*, sendo responsável por fornecer apoio ao consumo das informações contextuais provenientes da infraestrutura de software, podendo ainda fornecer suporte ao desenvolvimento de novas aplicações que possam vir a utilizar a infraestrutura.

Segundo (SOUSA, 2019) em um típico cenário de execução do INavigS as ações são executadas da seguinte forma:

1. Por todo o período em que o *smartphone* esteja fazendo uso da infraestrutura, periodicamente se realiza o escaneamento com o intuito de descobrir beacons próximos a sua localização e enviar tais dados ao *Context Manager*.
2. O *Context Manager* recebe os dados advindos do smartphone e os armazena em uma base de dados NoSQL, após isso os dados são disponibilizados para as camadas superiores.
3. Após o recebimento da solicitação de assistência de rota, uma rota é calculada e fornecida pelo *Context Reasoner*, tendo em vista o perfil fornecido do usuário (possui ou não alguma deficiência motora), passando a monitorar seus dados de contexto a fim de determinar sua localização dentro do prédio.

4. Em seguida, uma mensagem de configuração é enviada à aplicação cliente, caso o passo anterior seja realizado com sucesso.
5. O *Context Reasoner* passa a realizar o monitoramento dos dados armazenados no *Context Manager*, procurando identificar eventuais mudanças de posição do usuário, com o intuito de atualizar sua posição no espaço.
6. Caso mudanças contextuais ocorram, o *Context Reasoner* realiza um novo cálculo de rota, resultando em uma nova instrução passada ao usuário por meio do *Context Consumer*.
7. Os passos acima são repetidos até que o usuário alcance seu destino ou cancele a solicitação de assistência de rota.

2.6.2 Representação do ambiente indoor utilizada pelo INavigS

No que diz respeito a representação do modelo espacial, ela se dá através de um modelo espacial simbólico hierárquico, conforme os níveis descritos pela Figura 2.12. O espaço é dividido em três níveis: *Building*, onde se encontram as informações que descrevem o prédio (Nome, endereço). *BuildingElements* para descrever as entradas, saídas, hall, banheiros, corredores e escadas que constituem o prédio. A nível de *Beacons* e *Links* é demonstrado a existência dos enlaces de comunicação entre os beacons existentes no ambiente indoor. Cada beacon contém as informações de identificação *Universally Unique Identifier* (UUID), posição (latitude e longitude). Quanto aos *Links*, o atributo peso de cada Link tem seu valor descrito em metros, que correspondem a distância entre os beacons *source* e *target*.

A visão do modelo espacial utilizado pelo INavigS a nível de sensores, pode ser observado através da Figura 2.13, na qual os valores P0, P1 e P2 são representações dos beacons que estão posicionados de forma estática no ambiente, sendo que as coordenadas de todos os pontos mapeados são cruciais para que ocorra a realização do cálculo de rota. Para isso, primeiramente se obtém as coordenadas geográficas da posição inicial (P0) bem

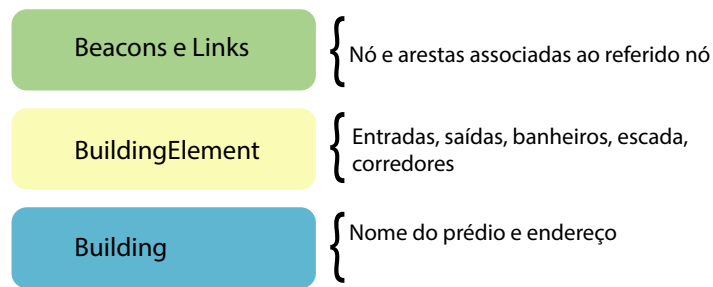


Figura 2.12 – Modelo Hierárquico do Espaço Indoor.

como, sua representação semântica (sala de aula 04, sala de coordenação de computação e etc.). Posteriormente são obtidos os valores referentes ao azimute representado pelos ângulos (α e β), e por último se obtém o valor das distâncias $d1$ e $d2$ entre os beacons posicionados no ambiente ($P1$ e $P2$).

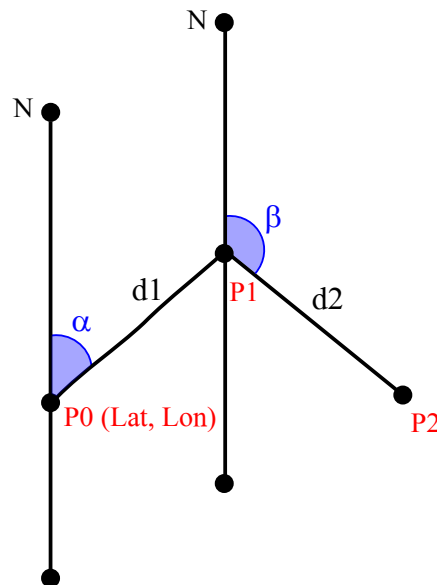


Figura 2.13 – Construção do modelo espacial de (MEDEIROS, 2018).

Ao fazer uso dessa abordagem para elaboração dos modelos de representação do espaço indoor, faz com que a adoção de um modelo de referência no formato CAD torne-se desnecessária. Além disso, possibilita a facilidade de se representar um espaço indoor, mesmo em ocasiões em que não haja uma planta baixa disponível, contendo as informações arquiteturais do prédio para nortear o processo de desenvolvimento do modelo

de representação, assumindo que o usuário tenha conhecimento acerca do ambiente e já o tenha explorado.

Uma vez que o modelo esteja construído, sua topologia é organizada na forma de um grafo, no qual beacons e a distância entre eles medida em metros representam nós e arestas respectivamente. Após a construção da topologia, os cálculos de rota podem ser realizados e a obtenção dos resultados é realizada por meio do algoritmo de Dijkstra para a obtenção do menor caminho entre dois pontos distintos no grafo que representa o modelo do espaço indoor. E por fim, as informações são armazenadas em um banco de dados relacional PostgreSQL.

2.7 Considerações Finais

Ao longo desse capítulo são abordados diversos conceitos fundamentais para a condução desse trabalho. A MDE, MDA e principalmente DSL e transformações entre modelos foram explorados, buscando a compreensão de como o desenvolvimento desse trabalho está atrelado a esse paradigma de desenvolvimento. Demonstrando ainda os benefícios advindos da utilização do modelo e da utilização de uma linguagem de modelagem, que forneça apenas conceitos englobados pelo domínio no qual se pretende desenvolver artefatos. Além dos requisitos necessários para que se possa desenvolver tal ferramenta.

No capítulo seguinte, é apresentado alguns dos trabalhos relacionados encontrados na literatura, a fim de identificar a utilização de DSL e dos princípios da MDE em alguma das etapas do processo de elaboração do modelo representação do espaço indoor.

3 Trabalhos Relacionados

Na busca por trabalhos que pudessem ter relação com a presente dissertação, procurou-se identificar trabalhos anteriores que apresentassem a mesma abordagem utilizada nesta pesquisa. Esse processo foi conduzido por uma revisão narrativa da literatura com o objetivo de coletar todas as informações relacionadas estudo. Sendo que a busca se iniciou por trabalhos (artigos, teses, dissertações e livros) ligados a computação, na qual não foram encontrados resultados de propostas similares, o que acarretou pela busca na área de cartografia pelo fato do assunto de domínio se apresentar atrelado a computação.

Apesar das buscas, não foram encontrados indícios de trabalhos relatos em que utilizasse os princípios da engenharia dirigida por modelos e/ou DSL, voltados ao desenvolvimento de um modelo de representação do espaço indoor. Embora não haja abordagens em comum a este projeto de dissertação, alguns trabalhos encontrados permitem a elaboração do modelo de representação do espaço indoor através de outras abordagens.

Diversas abordagens para elaboração do mapa do ambiente indoor que fornecem suporte para as aplicações IPS podem ser encontradas na literatura. Uma delas é baseada na utilização de sensores, câmeras e outros equipamentos que permitem realizar o mapeamento, a medida que o ambiente é percorrido, como proposto em (WEN et al., 2016) que realiza o mapeamento por meio uma mochila contendo três sensores lasers, um horizontal e dois verticais e um *Inertial Measurement Unit* (IMU)¹. O ambiente é percorrido enquanto o sistema escaneia todo o percurso para a realização do mapeamento, os dados coletados por meio dos sensores e IMU são processados com o auxílio do algoritmo SLAM resultando em dois mapas, um bidimensional e outro tridimensional.

¹Sensor composto tradicionalmente por giroscópio, acelerômetro que auxiliam na determinação da velocidade angular e aceleração.

Outras propostas similares que apresentam a utilização de equipamentos de hardware (sensores, lasers, câmeras, robôs) para mapear o ambiente podem ser encontradas em (Liao et al., 2019; Chan et al., 2018). Já no caso proposto em (Endo et al., 2017), o foco é na utilização do mapa combinado com o uso de sensores (câmera) para detecção de obstáculos e fornecer mecanismos de navegação para pessoas com deficiência motora .

E a partir disso, surgem novas iniciativas que visam obter esse mapeamento sem que haja um custo elevado que possa inviabilizar sua adoção. Como é o caso da utilização de mecanismos de colaboração, para que o mapeamento possa ser obtido a partir de dados oriundos de diferentes usuários, que exploram o ambiente e fornecem essas informações através de *Volunteered Geographic Information* (VGI) (informação geográfica voluntária) para o processo de construção do mapa do ambiente indoor. E uma dessas iniciativas é o projeto OSM, em que uma comunidade formada por especialistas em cartografia e pessoas de qualquer outra área possam contribuir para o mapeamento dos ambientes indoor gratuitamente. Abordagem utilizada por (FADLI et al., 2018) no qual o mapa no modelo IFC contendo os elementos arquiteturais que compõem o prédio é sobreposto sobre o mapa geográfico, e a partir dessas sobreposição é possível desenhar os elementos e construir as relações por meio de um editor JOSM, que ao ser integrado aos dados já obtidos do OSM permite gerar um mapa no formato CityGML da torre da agência de notícias do Qatar, a fim de fornecer informações de navegação em situações de evacuação de emergência. Outras concepções propõem o modelo obtido a partir do OSM com a utilização de tags customizadas, com o intuito de otimizar o sistema de marcação que contém a representação do espaço, que podem ser reduzidas como no caso proposto por (GRAICHEN et al., 2017) ou aprimoradas como proposto em (WANG; NIU, 2018). Há outra alternativa em que dados podem vir de várias pessoas, a fim de elaborar e identificar as áreas navegáveis em um ambiente indoor. Tais técnicas visam obter informação do ambiente por meio de pessoas e sensores (como os provenientes dos dispositivos móveis dos usuários), atuando juntos para promover a construção do mapa como proposto em (GAO et al., 2018). E a partir dos dados de rastreabilidade gerado do ambiente, pode estimar por exemplo um conjunto de combinações formadas

pelos polígonos e construir uma gramática estrutural baseada nos traços gerados pelos pedestres e ao combinar fragmento dos traços com a gramática definida se obtém o mapa do ambiente indoor, como proposto em (HU et al., 2017).

Tratando-se de utilizar um arquivo CAD, como mecanismo de entrada ou guia da obtenção do modelo de representação do espaço indoor, algumas abordagens têm proposto soluções próprias como, como no caso proposto por (LIN et al., 2018; YANG; WORBOYS, 2015), que apresentam um método de extração de dados para geração de mapas topológicos, a partir de uma dada planta arquitetural que passa por uma etapa de pré-processamento destinada a retirada de elementos que não são de importância para a obtenção do mapa (tais como hidrante e colunas), focando na identificação de salas, paredes e portas para a obtenção do mapa por identificação de pontos, linhas e arcos (esse último destinado a identificação das portas). Abordagem similar foi utilizada também por (SRIVASTAVA et al., 2018), que tem como proposta a extração do modelo de representação do espaço indoor a partir de uma planta desenhada em CAD, onde todas as informações extraídas são convertidas para o formato IndoorGML contendo geometria, topologia e informações semânticas. Sendo que as primitivas consideradas no trabalho estão relacionadas somente as paredes, textos (informações acerca da identificação do elemento arquitetural) e portas. O trabalho leva em consideração somente polígonos retilíneos (linhas curvas estão fora do escopo do trabalho). O processo de extração se baseia na simbologia utilizada para representar elementos arquiteturais usados no desenho CAD.

A maioria dos trabalhos focam na utilização de um processo de extração dos dados a partir do arquivo CAD contendo a planta arquitetural do ambiente indoor, ou de forma semiautomatizada (LIN et al., 2018; YANG; WORBOYS, 2015) ou de forma manual e repetitiva, editando ponto a ponto as primitivas utilizadas no modelo de representação (WANG; NIU, 2018), o que acaba se tornando uma tarefa propensa a erros. Ou ainda, apresentam um custo elevado de implantação devido a fatores logísticos (WEN et al., 2016; Liao et al., 2019).

A fim de traçar uma identificação das características entre os trabalhos correlatos.

Foi elaborada a Tabela 3.1 na qual apresenta um comparativo entre as diferentes abordagens, nos quais são levantadas as seguintes questões:

Q01: Há um modelo de representação do ambiente?

Q02: É empregado na navegação?

Q03: É dependente de um modelo arquitetural CAD?

Q04: Fornecem meio de construção para usuários finais?

Q05: Utiliza os princípios da MDE, ou fornece uma DSL visual para o processo de modelagem?

Os casos omissos durante o preenchimento ocorrem em virtude da ausência de informações no texto que possam vir a responder as questões levantadas.

Tabela 3.1 – Tabela comparativa entre os trabalhos correlatos.

Trabalho	Q01	Q02	Q03	Q04	Q06
Implantação de SLAM indoor 2D laser em um robô móvel baseado em raspberry pi (Liao et al., 2019).	✓	✗	✗	✗	✗
Uma nova abordagem de mapeamento indoor amigável para o openstreetmap (GRAICHEN et al., 2017).	✓	✓	✓		✗
Um sistema de mochila indoor para mapeamento 2D e 3D de interiores de edifícios (WEN et al., 2016).	✓	✓	✗	✗	✗
Localização interna 2d robusta por meio de laser SLAM e fusão visual SLAM (Chan et al., 2018).	✓	✗	✗	✗	✗
Geração de grafos de navegação para espaço indoor (YANG; WORBOYS, 2015).	✓	✓	✓		✗
Um modelo de dados para usar o openstreetmap para integrar planejamento de rotas internas e externas (WANG; NIU, 2018).	✓	✓	✓	✓	✗
Estendendo o mapeamento de ambientes indoor openstreet para modelo de edificação 3D CityGML navegável: avaliação de resposta em emergência (FADLI et al., 2018).	✓	✓	✓	✓	✗
Construção de mapa interno por meio de crowdsensing móvel (GAO et al., 2018).	✓	✓	✗		✗
Um método de geração de mapas topológicos baseado em grafo para localização interna (LIN et al., 2018).	✓	✓	✓		✗
Posicionamento interno e detecção de obstáculos para sistema de navegação para deficientes visuais baseado em LSD-SLAM (Endo et al., 2017).	✓	✓	✗	✗	✗
Para gerar dados indoorGML semanticamente ricos a partir de plantas arquitetônicas (SRIVASTAVA et al., 2018).	✓	✓	✓		✗
Um framework conceitual para mapeamento indoor por utilizar gramáticas (HU et al., 2017).	✓	✗	✗		✗

4 A Linguagem de modelagem IndoorDSL

Este capítulo tem o objetivo de apresentar a proposta abordada neste trabalho de dissertação. Em virtude da ausência de trabalhos na literatura que apresentem proposta de obtenção do mapa de representação do espaço indoor, com a abordagem orientada a modelos como elemento central no desenvolvimento, bem como fornecer a possibilidade de ferramentas de transformação para os modelos desenvolvidos.

Nas seções seguintes, serão abordadas as etapas no processo de construção da linguagem de modelagem IndoorDSL, evidenciando as ferramentas que foram empregadas para sua obtenção.

4.1 Visão geral

O desenvolvimento teve como ponto de partida a busca pela construção de uma linguagem de modelagem específica, voltada ao desenvolvimento de modelos de representação do espaço indoor utilizados pela infraestrutura de software INavigS. Até a conclusão do projeto, algumas etapas devem ser seguidas para que se obtenha êxito na elaboração da linguagem. Segundo Deursen et al. (2000), a metodologia de desenvolvimento de uma DSL devem compreender tipicamente os seguintes passos:

Análise: uma vez que o problema tenha sido identificado, no caso do INavigS, a ausência de uma linguagem com um alto nível de abstração, que pudesse fornecer suporte ao desenvolvimento dos modelos de representação do espaço indoor utilizados pela infraestrutura de software. A fase de análise é responsável pela condução da busca por todas as informações relevantes pertinentes ao domínio de estudo, em que a condução da tarefa de coleta pode ser atribuída ao expert do domínio em virtude de possuir uma maior familiaridade com todas as características pertinentes ao

negócio. Nessas ocasiões é dito que o processo de análise se deu de forma ad-hoc.

Implementação: nessa fase, ocorre o processo de construção da linguagem de modelagem que servirá para atender a um propósito específico. No caso deste projeto de dissertação a concepção do metamodelo a partir das ontologias que constituem o modelo do espaço indoor identificadas no processo anterior, bem como a construção dos elementos gráficos que deverão representar tais conceitos. Além das definições constituintes do modelo de ligação entre a metamodelo e modelo, que serão responsáveis por conter a gramática responsável por garantir a corretude das notações gráficas.

Uso: consiste na criação de modelos de representação do espaço indoor por meio da DSL desenvolvida, engloba também a realização das transformações que porventura são permitidas sobre o modelo.

Sendo assim, a metamodelagem teve como ponto de partida a análise da representação do espaço indoor utilizada na dissertação de (MEDEIROS, 2018), e a partir dessa análise, deu-se início ao desenvolvimento de uma ferramenta que pudesse fornecer um ambiente que permita a criação de modelos e realização de transformações, para que possam ser utilizados pelo INavigS. Com a análise do código e dados contidos no banco de dados utilizado pela infraestrutura de software, foram identificados os conceitos e propriedades contidos Tabela 4.1:

Para permitir que o modelo de representação do espaço fosse criado em mais alto nível, algumas questões chave foram elencadas para nortear o processo de desenvolvimento da ferramenta, sendo elas:

Orientada ao modelo: Todas as ocorrências de instâncias feitas pela ferramenta de modelagem devem corresponder aos elementos identificados na etapa de metamodelagem. Sendo assim, correspondem diretamente aos elementos identificados a partir da análise do domínio.

Tabela 4.1 – Elementos extraídos após a análise da plataforma INavigS.

Classe	Descrição	Atributos
Address	Diz respeito a identificação de um ponto existente em uma cidade, traz as informações que residem a nível de prédio descritas no modelo de representação utilizado pelo INavigS descritos no Capítulo 3.	idAddress city state zipCode number complement neighborhood
Build	Essa é a informação do prédio que podem ser inseridos um determinado endereço.	idBuild idAddress name
BuildElement	Elementos que compõe o prédio de forma genérica podendo ser uma sala, corredor, banheiro e etc.	idBuildElement idBuild description name type accessible vertical
Beacon	Descreve os sensores bluetooth que poderão ser espalhados pelo ambiente.	idBeacon idBuildElement code uuid description latitude longitude
Link	Esse elemento representa a ligação entre dois sensores (beacons) podendo haver ligação entre um ou mais elementos.	idBeacon1 idBeacon2 idLink weight azimuth

Simplicidade: Por compreender aspectos relacionados especificamente ao domínio, a linguagem de modelagem apresenta representação de poucos objetos, o que facilita seu aprendizado no decorrer do uso da ferramenta.

Conformidade: Em virtude do escopo do trabalho ser voltado aos modelos utilizados pela infraestrutura de software INavigS, o resultado da transformação do modelo gerado deve permitir que possa ser utilizado por ela, servindo como entrada de informações (modelo) para o INavigS.

Confiabilidade e precisão: A confiabilidade na geração dos modelos foi alcançada em virtude da consistência com as informações especificadas no metamodelo e na definição das notações gráficas utilizadas para elaboração dos modelos. Permitindo

que possam ser validadas e assim corrigir possíveis erros que possam ocorrer a partir da construção do modelo.

Para a elaboração da linguagem de modelagem voltada ao desenvolvimento de mapas de representação do espaço indoor utilizados pela infraestrutura de software INavigS foram considerados dois grandes passos: definir os requisitos que a linguagem deve possuir e as decisões a serem tomadas para as fases de definição de projeto e elaboração da linguagem.

4.2 Requisitos e decisões de projeto

A seguir são, apresentados alguns requisitos que foram almeçados pela linguagem de modelagem, bem como as decisões de projeto que culminaram nesse trabalho de dissertação de mestrado.

4.2.1 Requisitos do projeto

- R1:** A DSL deve prover uma representação gráfica contendo todos os elementos que fazem parte do domínio.
- R2:** Tornar o processo de elaboração do mapa de representação do espaço indoor mais simples, em relação a abordagem tradicionalmente utilizada. E fornecer uma melhor compreensão acerca do sistema modelado.
- R3:** Permitir que o metamodelo possa ser estendido, para que possa vir a fornecer suporte a algum IPS que possuam similaridades com o modelo utilizado pelo INavigS.
- R4:** A DSL deve fornecer suporte a transformação para outros modelos.
- R5:** Tanto a utilização das ferramentas para elaboração da DSL, quanto a sua disponibilização devem ser gratuitas.

Em se tratando das decisões de projetos tomadas para a elaboração da DSL, são as seguintes:

- D1:** Linguagem de modelagem gráfica. Com a finalidade de permitir que conceitos do domínio possam ser representados por ilustrações e assim abstrair aspectos complexos da geração do modelo. Outro fator determinante na escolha da linguagem gráfica, foi a familiarização com as ferramentas utilizadas no processo de definição da linguagem proposta neste projeto de dissertação.
- D2:** Geração dos artefatos em Java. Os artefatos resultantes da transformação dos modelos concebidos por meio da linguagem de modelagem tem Java como principal linguagem de programação, devido ao fato de correspondência entre os artefatos gerados e o conjunto de classes *Data Access Object* (DAO) existentes na infraestrutura de software INavigS (escopo deste trabalho).

4.3 Sintaxe Abstrata

A MDE possui o desenvolvimento focado na utilização de modelos, permitindo se representar sistemas por meio de modelos. O primeiro passo para que se possa especificar a criação de vários modelos é através da elevação do nível de abstração com a especificação de um metamodelo. Nesse caso, englobando conceitos advindos das regras de negócio estabelecida pelo INavigS para que se pudesse criar a DSL.

A concepção do metamodelo da linguagem de modelagem IndoorDSL ocorreu através da ferramenta EMF presente na instalação do OBeo, na qual se permite elaborar metamodelos representados através de arquivos *Ecore* contendo classes, atributos, relações e tipos de dados utilizados para compor os conceitos e relações existentes no metamodelo e oriundos do domínio em questão. O metamodelo obtido pode ser observado na Figura 4.1.

As classes contidas no metamodelo foram obtidas a partir da análise do modelo de representação do espaço indoor utilizado no INavigS. O metamodelo tem a classe

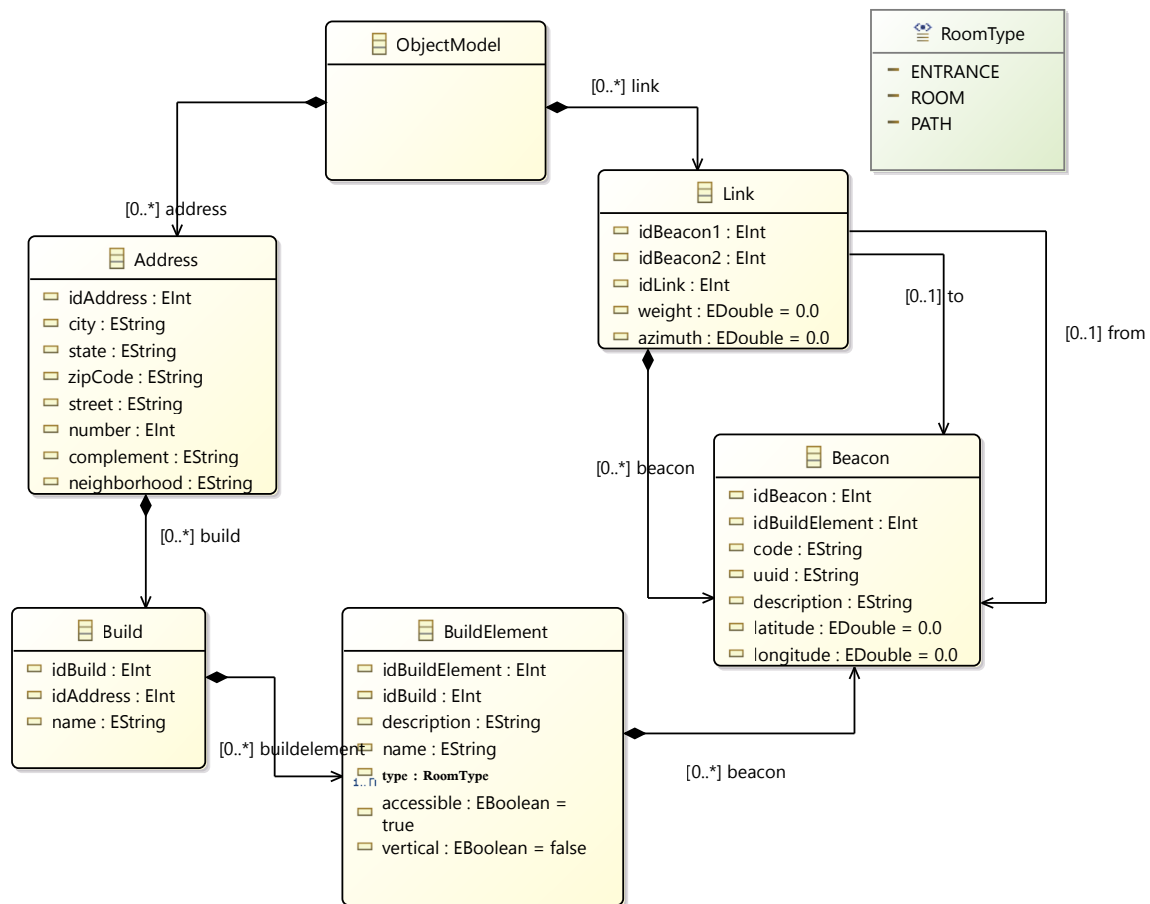


Figura 4.1 – Metamodelo da linguagem de modelagem IndoorDSL.

ObjectModel como ponto inicial pra a criação de um modelo, sendo que a partir dela se pode definir a classe “Address”, que contém as informações a nível de prédio, e a “Link” que é responsável pela interligação entre um ou mais beacons que estejam espalhados pelo ambiente (a nível de sensores).

Após a inserção do “Address”, é possível inserir a classe “Build” que faz referência aos prédios que podem ser inseridos no endereço já especificado. Depois que houver um prédio inserido se pode definir os elementos que o constituem, representados pela classe “BuildElement” no metamodelo. Os beacons poderão ser atribuídos ao “BuildElement” a fim de criar a camada de sensores sobre a camada de elementos arquiteturais, sendo que cada “BuildElement” pode receber um ou mais “Beacons”. Por último é possível definir o “Link” entre “Beacons”, onde um “Beacon” pode estabelecer uma ligação com um ou mais

“Beacons”.

4.4 Sintaxe Concreta

Para que seja possível a manipulação dos elementos que definem o modelo e esteja em conformidade com o que foi especificado no metamodelo (sintaxe abstrata), faz-se necessário uma linguagem que contenha a simbologia de tais elementos. De modo que, possa permitir a criação de notações gráficas (gramática) pelas quais se possa criar modelos. A gramática visual vai ser responsável por descrever a estrutura dos elementos contidos no diagrama de forma independente da imagem/símbolo utilizado, por fornecer uma hierarquia na qual apresenta a descrição das relações entre os elementos. Um elemento pode conter outros elementos (subelementos), e as relações que podem ocorrer entre elementos que podem ou não ser ligados a outros elementos.

Toda sua elaboração ocorreu através da ferramenta Sirius¹ que fornece a possibilidade de utilizar um metamodelo de referência para uma notação gráfica e criação de workbench de modelagem no qual se utiliza tais notações. Além disso, o Sirius permite que usuários possam criar, editar e visualizar modelos EMF com suporte a diferentes modelos de editores para os modelos, que podem ser editados através de diagramas, tabelas, ou árvores (ARRUDA, 2019). A escolha da ferramenta ocorreu em virtude da familiarização com ambiente. Embora necessite de um maior esforço de aprendizado (*Learnability*) para criar uma DSL, ainda assim possui a habilidade de representar elementos gráficos de maneira vasta e com diferentes estilos (*Graphical Completeness*), conforme evidenciado no comparativo proposto no trabalho realizado por (RIBEIRO et al., 2016). O papel do Sirius, no desenvolvimento da DSL abordada nesse projeto de dissertação, é de fornecer a possibilidade de realizar a ligação entre os elementos contidos na metaclassa com suas correspondentes ilustrações, na sua representação gráfica por meio de um modelo de ligação entre metamodelo e modelo, tal papel é evidenciado conforme ilustrado na Figura 4.2.

¹<https://www.eclipse.org/sirius/>

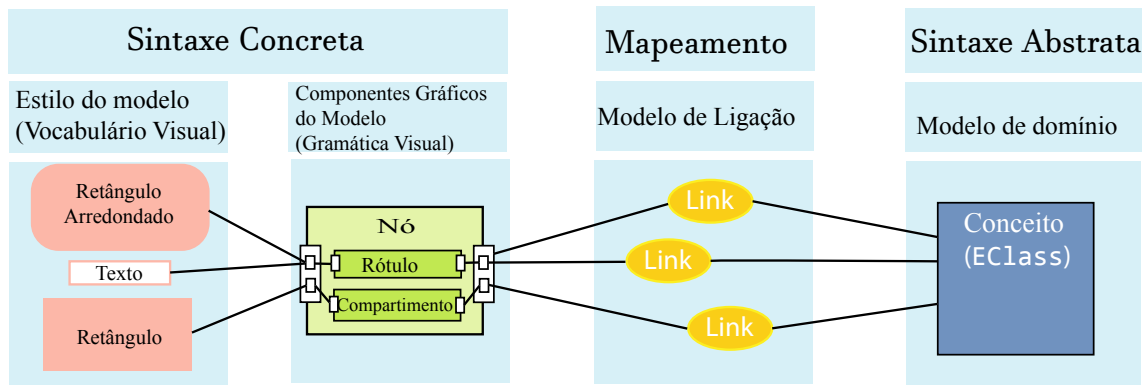


Figura 4.2 – Partindo de metamodelos até a interface de diagramas do usuário. Adaptação de (KOUHEN, 2013).

Para que seja criada uma representação gráfica é necessário a criação de um projeto Sirius contendo a “Viewpoint Specification Model” no formato de arquivo “*.odesing”, cuja finalidade é a definição e visualização de uma metaclassa na forma de aspecto gráfico, no qual terá o papel de modelo de ligação entre a interface final e a metaclassa. Já as definições dos elementos contidos na metaclassa para o ponto de visão oferecido pelo Sirius é realizada por expressões escritas na linguagem AQL². Um *viewpoint* pode ser observado na Figura 4.3 no qual é possível observar a hierarquia do projeto da linguagem IndoorDSL.

Todos os elementos contidos no metamodelo devem ser definidos no *viewpoint*, são eles: Address, Build, BuildElement, Beacon e Link. O Sirius permite definir diferentes elementos gráficos que podem vir a compor o vocabulário de uma linguagem visual (ECLIPSEORG, 2020a), dentre todos os elementos, somente quatro foram utilizados no desenvolvimento deste trabalho, nos quais são:

Nó: utilizado para a representação de um elemento do modelo, não possuindo qualquer outro sub elemento, ou referência a outro objeto no qual não seja especificado inicialmente.

Contêiner: um container permite que se possa inserir sub elementos dentro de sua

²<https://www.eclipse.org/acceleo/documentation/>

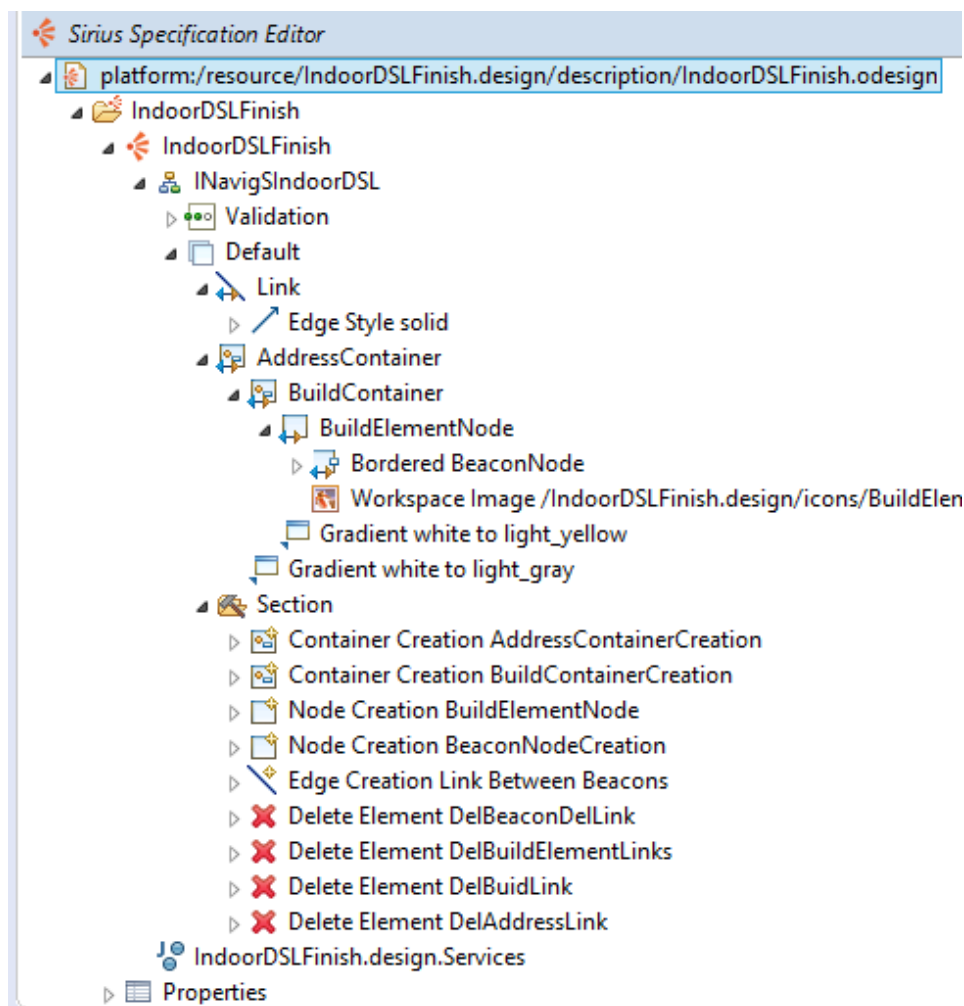


Figura 4.3 – Sirius Viewpoint da linguagem IndoorDSL.

estrutura que podem ser nós, relações ou até mesmo outros containers de forma recursiva.

Nó de borda: consiste em um tipo de nó que é mapeado para que fique sempre as margens de um nó predefinido, em vez de aparecer diretamente no diagrama sua definição é atrelada a um nó em específico.

Relação: elemento que permite visualizar a relação entre dois outros elementos através de um elo que pode interligá-los.

O processo de desenvolvimento da parte gráfica corresponde a linguagem IndoorDSL se deu a partir do fluxo de trabalho exposto na Figura 4.4 apresentando

um conjunto de passos que resultaram na linguagem de modelagem. Sendo descritos individualmente e compreendendo o conjunto de ações realizadas ao longo de cada passo.

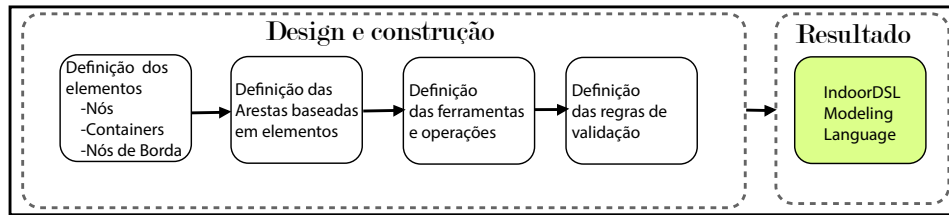


Figura 4.4 – Etapas do processo de construção gráfica da linguagem IndoorDSL

Definição dos elementos

A definição do nó compreende a realização do mapeamento que descreve qual elemento do modelo deverá ser visualizado na linguagem de modelagem gráfica. Em acordo com o “viewpoint” abordado, apenas o elemento `BuildElementNode` é caracterizado como sendo um nó (Figura 4.5a). Sua definição no Sirius segue a seguinte especificação de propriedades. Os elementos `AddressContainer` (Figura 4.5b) e `BuildContainer` (Figura 4.5c) são utilizados para definir os containers existentes na composição do vocabulário gráfico. Além disso, há um nó de borda `Bordered BeaconNode` (Figura 4.5d) que faz alusão aos beacons e sendo mapeado de forma que sempre que houver sua ocorrência, essa esteja vinculada a um `BuildElementNode`.

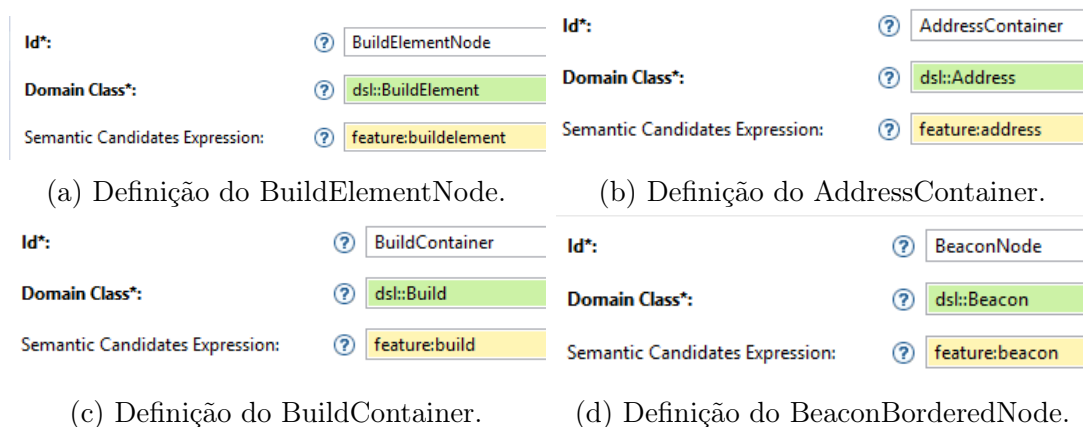


Figura 4.5 – Definição dos elementos gráficos contidos na linguagem IndoorDSL.

Definição dos elementos baseados em arestas

No que diz respeito ao elemento `Link`, este é definido como sendo um elemento visual do tipo relação sendo um *Element-Based Edges*, que é utilizado para representar relações quando há um elemento no metamodelo como sendo a própria relação (em vez de uma referência) sua definição gráfica é realizada pela configuração expressa na Figura 4.6.

Id*:	<input type="text" value="Link"/>
Domain Class*:	<input type="text" value="dsl::Link"/>
Source Mapping*:	<input type="text" value="Bordered BeaconNode"/>
Source Finder Expression:	<input type="text" value="feature:from"/>
Target Mapping*:	<input type="text" value="Bordered BeaconNode"/>
Target Finder Expression*:	<input type="text" value="feature:to"/>

Figura 4.6 – Definição visual do elemento `Link`.






Definição das Ferramentas/Paleta e operações

Quanto aos elementos responsáveis por compor a linguagem graficamente (sintaxe concreta), que possibilitarão a elaboração de notações gráficas de forma a representar os conceitos definidos no metamodelo, podem ser encontrados na Tabela 4.2. Tais ilustrações são fornecidas como ferramentas da linguagem, para que se permita criar os modelos a partir delas.

Regras de validação

É comum usuários não terem a devida atenção a explicação do uso de um software ou ao seu manual, em vez disso preferem explorar a aplicação e aprenderem a partir dos erros. A partir disso, buscou-se ser o mais claro e objetivo possível nas mensagens

Tabela 4.2 – Elementos utilizado na notação gráfica

Sintaxe Abstrata	Sintaxe Concreta/Ícone	Descrição
Address		Declaração de um objeto do tipo endereço
Build		Utilizado para declarar um prédio
BuildElement		Declaração de um elemento contido em um prédio
Beacon		Declaração do sensor beacon
Link		Enlace entre dois ou mais beacons

resultantes da aplicação das regras de validação do uso da linguagem. Para isso, as regras foram aplicadas conforme especificada em (ROSA, 2017).

- Regras de validação são expressas de maneira textual.
- As regras se referem a parte ou partes do modelo que possam resultar em erros.
- As regras são específicas o bastante para que usuários possam entender o feedback.
- As regras são integradas a linguagem a fim de prover erros e sugestões de possíveis soluções.

Ao definir as regras de validação semântica através do Sirius para tratar eventuais erros no processo de modelagem, é possível especificar três tipos distintos mensagens, que podem ser um *Error*, *Information* ou *Warning*. Todo o processo de identificação do erro é avaliado por expressões escritas em AQL, podendo por exemplo, auxiliar na identificação de campos que deixaram de ser preenchidos durante a modelagem. A realização do processo de validação é feita na área de modelagem, no qual o usuário deve clicar na área vazia com o botão direito do *mouse* e escolher a opção “Validate Diagram” como por exemplo, o uso das expressões para a verificação dos campos `Address.city` (Expressão 4.1), `Address.state` (Expressão 4.2) e `Build.type` (Expressão 4.3).

$$aql : self.city \rightarrow isEmpty() \quad \text{ou} \quad aql : self.city \neq null \quad (4.1)$$

$$aql : self.state \rightarrow isEmpty() \quad (4.2)$$

$$aql : self.type \neq null \quad (4.3)$$

Algumas outras regras podem ser definidas para que englobe todas as possíveis ocorrências de erros. Como no caso da verificação do erro semântico, no qual um beacon pode estabelecer um `Link` com outro beacon, mas não se permite que um beacon seja tanto emissor quanto receptor em um mesmo `Link`. Tal regra de validação é alcançada através do uso da Expressão 4.4, o resultado do seu processamento pode ser observado na Figura 4.7.

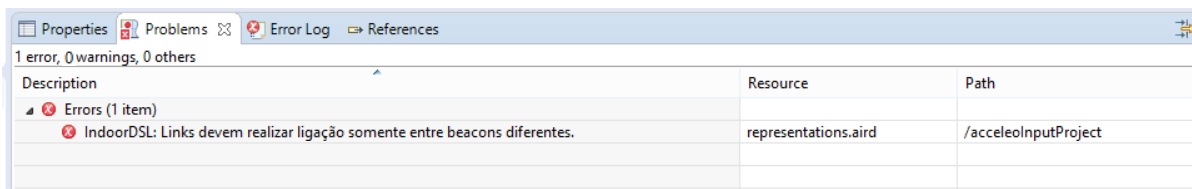
$$aql : self.from \neq self.to \quad (4.4)$$


Figura 4.7 – Erro resultante da expressão que avalia a validade do `Link`.

A existência de inconsistências pode ser identificada no momento da modelagem, por meio da validação dos campos destinados as propriedades do objeto inserido. No uso da Expressão 4.5, que tem a finalidade de não permitir que um prédio receba o nome de um prédio já existente, por exemplo. O processamento resultante da expressão resulta em um erro durante o processo de modelagem, conforme demonstrado na Figura 4.7.

$$aql : not self.siblings() \rightarrow filter(dsl::Build) \rightarrow collect(value | value.name) \rightarrow includes(self.name) \quad (4.5)$$



Figura 4.8 – Erro resultante ao definir um nome já existente a um prédio.

Resultado

O resultado obtido através das etapas e definições por meio do Sirius foi um ambiente de desenvolvimento gráfico, que permite criar modelos de representação do espaço indoor a partir de componentes gráficos que estão em acordo com o domínio de estudo. Tarefa obtida a partir da interface fornecida pela Figura 4.9, no qual ilustra o ambiente de modelagem oferecido pela linguagem IndoorDSL. Consistindo de três áreas com finalidades distintas, sendo elas:

1-Área de modelagem gráfica: área destinada a modelagem na qual usuários podem criar seus próprios modelos livremente. Permitindo que se insira os elementos gráficos contidos que estejam constando na paleta.

2-Propriedades: nessa área, é possível definir valores para os atributos contidos nos elementos gráficos inseridos no modelo por meio de um formulário. O aspecto do formulário de propriedades muda de acordo com o elemento selecionado na área de edição, sempre mostrando os campos que podem ser editados em relação ao referido elemento.

3-Paleta: a paleta contém todos os elementos gráficos, que podem ser utilizados para representar um modelo *conformsTo* metamodelo. Estão ordenados na forma cronológica de inserção na área de modelagem a fim de facilitar a compreensão do uso da ferramenta.

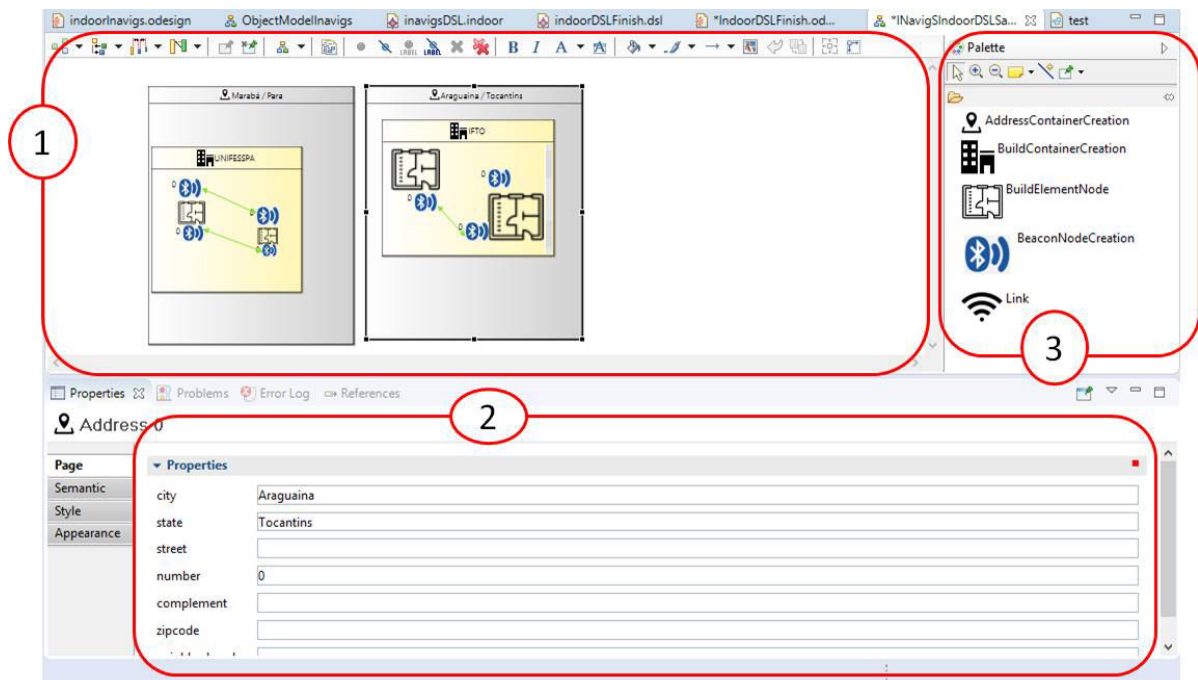


Figura 4.9 – *Screenshot* da interface de modelagem da linguagem IndoorDSL

4.5 Transformações

Após finalizado o processo de elaboração dos modelos de representação do espaço indoor por meio da IndoorDSL, o passo seguinte é a realização da transformação do modelo em texto (transformação M2T), realizada por intermédio do Aceleo, e que permite especificar um template (arquivo *.mtl) responsável pela realização dessa transformação. O processo se dá a partir do template, que tem o papel de obter todos os elementos contidos no modelo de representação do espaço indoor criado na linguagem IndoorDSL. Que pode vir a compor o modelo de representação do espaço indoor utilizado pela infraestrutura de software do INavigs.

Uma visão acerca do processo de desenvolvimento dos mapas que o INavigS pode utilizar é vista na Figura 4.10. Onde o usuário por meio da interface gráfica oferecida pela IndoorDSL, pode desenvolver o modelo que contém o mapa utilizado pelo

INavigS, através da manipulação dos elementos gráficos que compõem a sintaxe concreta. Finalizado o modelo contendo o mapa do ambiente indoor, é possível gerar um artefato de código Java contendo todas as instâncias especificadas no modelo. Tais instâncias podem ser passadas diretamente aos métodos responsáveis pela inserção dos elementos que irão compor o mapa na base de dados do projeto do INavigS.

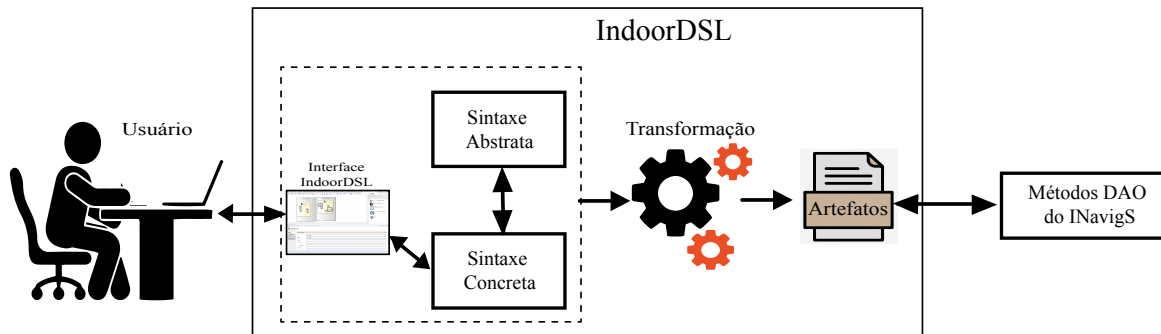


Figura 4.10 – Visão geral do processo de desenvolvimento dos mapas utilizados pelo INavigS.

O texto obtido com a transformação M2T é um artefato em Java contendo todas as instâncias definidas nos modelos. Devido ao fato do INavigS ter sido desenvolvido fazendo o emprego do padrão estrutural DAO, possuindo um conjunto de classes responsáveis pela manipulação da base de dados. Todas as instâncias podem ser passadas para os métodos DAO responsáveis pela realização dessa inserção na base de dados relacional, para que a infraestrutura possa popular a sua base de dados e assim, passar a ter o mapa de representação do ambiente indoor para um dado modelo. A Figura 4.11 ilustra o fragmento do template para a transformação das instâncias do objeto `BuildElement` bem como seu respectivo fragmento de código gerado, contendo as instâncias descritas no modelo. Todos os artefatos são provenientes do modelo demonstrado no Apêndice B.1. Devido ao fato de que os beacons são os elementos utilizados para prover a navegação na infraestrutura de software, os dados a nível de prédio do modelo não são tomados como exemplo de geração dos artefatos. No entanto, a íntegra do código gerado pode ser visto no Apêndice C.1.

Após a inserção dos elementos `BuildElement` que compõe o prédio é possível

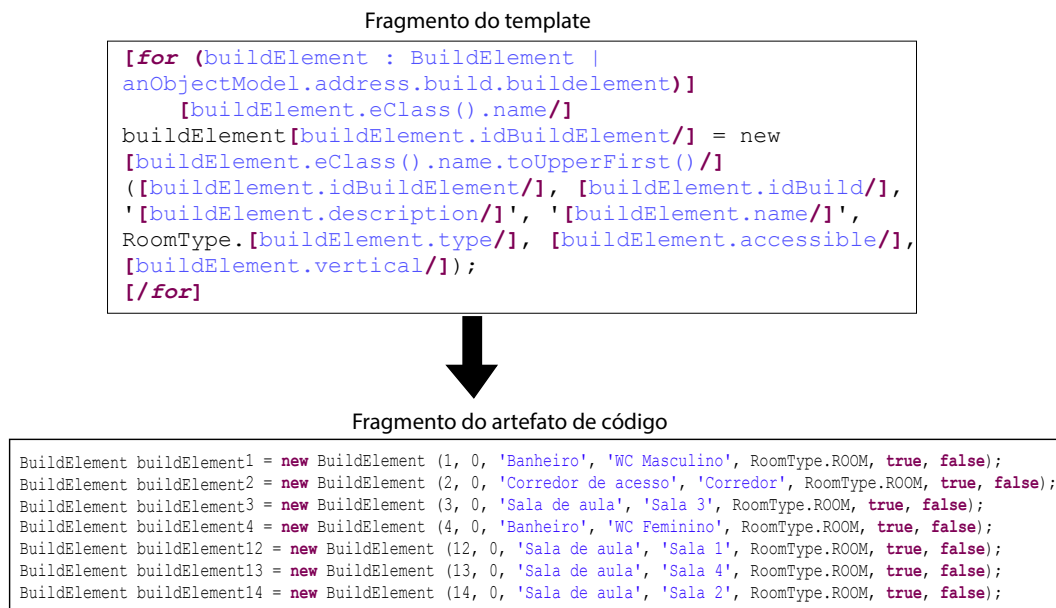


Figura 4.11 – Fragmento do template destinado a geração para BuildElement e o respectivo código.

posicionar os beacons nos elementos arquiteturais. Uma vez posicionados, há um fragmento no template destinado a transformar todos os elementos inseridos no modelo por meio da sintaxe concreta que especifica o elemento Beacon em instâncias de código Java, conforme pode ser visto na Figura 4.12 .

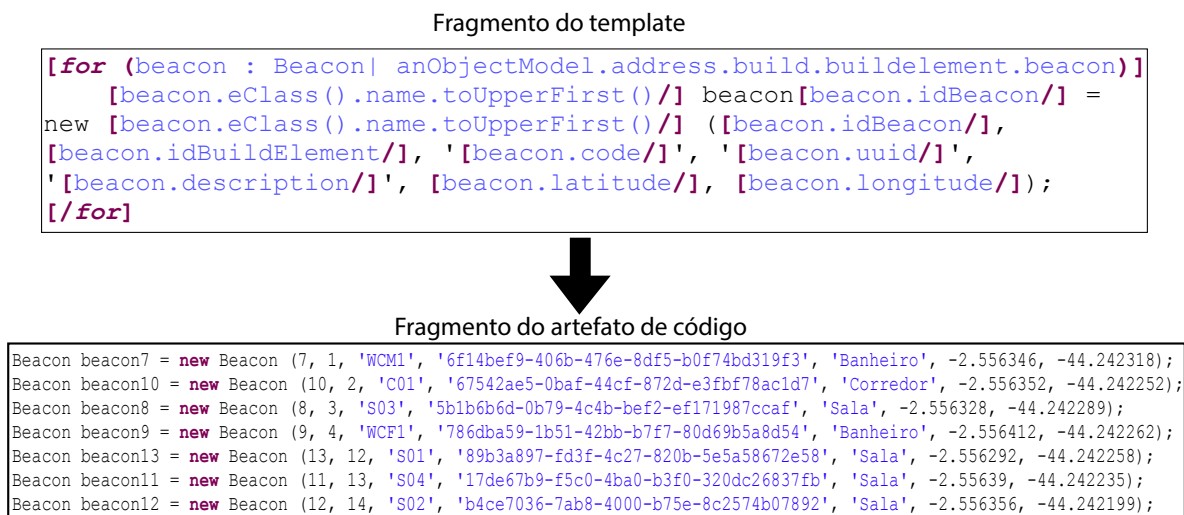


Figura 4.12 – Fragmento do template destinado a geração para Beacon e o respectivo código.

E por fim, é estabelecido os links entre os beacons para que possam formar o grafo que define a topologia utilizada na navegação do INavigS. O fragmento do template responsável por realizar as transformações das ligações entre beacons, definidas por meio da IndoorDSL é evidenciado pela Figura 4.13 que apresenta também, as instâncias resultantes das transformações.

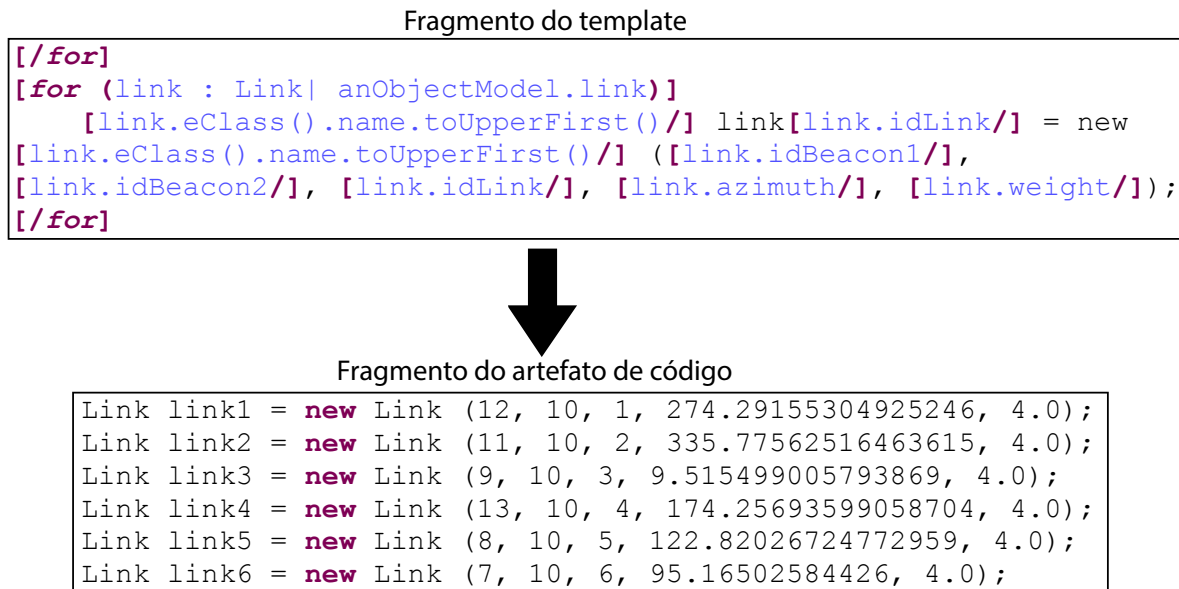


Figura 4.13 – Fragmento do template destinado a geração para Link e o respectivo código.

Ao final, é gerado um artefato de código contendo todas as instâncias dos elementos que compõe o mapa utilizado para que INavigS possa prover suas funcionalidades. Dessa forma, as classes DAO do INavigS podem acessar tais instâncias para a obtenção do mapa construído por meio da linguagem de modelagem IndoorDSL.

4.6 Considerações finais

Este capítulo apresentou o processo de concepção da linguagem de modelagem IndoorDSL, na qual tem por finalidade, o desenvolvimento de modelos de representação do espaço indoor (mapa) voltados a infraestrutura de software INavigS. Compreendendo a descrição do metamodelo (sintaxe abstrata) contendo os elementos extraídos do domínio

ao qual a linguagem é aplicada. Descreve ainda os elementos gráficos que integram o vocabulário da linguagem (sintaxe concreta) que deve estar em conformidade (*conformsTo*) o metamodelo anteriormente definido, bem como processo de definição da linguagem por meio das ferramentas escolhidas para a sua concepção.

Ao seu decorrer, o capítulo abordou o conjunto de regras de validação utilizadas, para assegurar que o usuário possa manter a completude do modelo, bem como eventuais erros e inconsistências, que podem ser geradas durante o processo de modelagem. Apresenta ainda, a interface fornecida para que se possa elaborar os modelos de representação do espaço indoor. E por fim, têm-se o processo de realização de transformação permitido pela linguagem de modelagem, responsável por converter os modelos manipulados por meio de sua interface gráfica em artefatos de código na linguagem Java, que é o produto resultante do processo de criação do mapa utilizado pelo INavigS.

5 Avaliação

No Capítulo 4, foi descrita uma linguagem de modelagem que permite a elaboração de um modelo de representação do espaço indoor, através de uma interface gráfica, capaz de fornecer auxílio no processo de validação do modelo desenvolvido por fornecer informações de inconsistências encontradas, que fazem com que o modelo não esteja completamente correto. Já neste capítulo, é apresentado a condução dos experimentos que visam identificar as medidas quantitativas e qualitativas a respeito do uso da ferramenta por parte dos usuários finais (leigos ou experts no domínio), da linguagem de modelagem apresentada neste trabalho de dissertação, com intuito de identificar sua eficácia.

5.1 Experimento

Nessa etapa do trabalho, buscou-se avaliar a ferramenta para identificar sua eficácia, verificar a percepção dos usuários em relação a linguagem, além de sua eficiência como uma linguagem de modelagem, que pudesse ser empregada no desenvolvimento de modelos de representação do espaço indoor. Tendo como base alguns aspectos que auxiliam no processo de validação, que segundo Diniz (2016), são separadas em quatro características nas quais um sistema deve apresentar, sendo capaz de ser:

Fácil: permitir que usuários leigos na criação de modelos de representação do espaço indoor possam desenvolver modelos simples e complexos por meio da DSL proposta.

Objetivo: Os modelos definidos por meio da linguagem de modelagem devem ser traduzidos corretamente em acordo, com o que é exigido pelo domínio.

Dinâmico: Usuários podem realizar transformações, convertendo seus modelos em artefatos, que possa ser utilizado pela infraestrutura de software, através de um processo automatizado.

Axiomática: A interface visual deve apresentar conceitos relacionados ao domínio, e ainda assim fornecer um ambiente compreensível e auto explicativo para diferentes grupos de usuário. Para que possa ser capaz de identificar os elementos, compreendê-los e assim, construir seus próprios modelos a partir da interação com a linguagem de modelagem.

5.1.1 Objetivos e Métricas

A realização de experimentos tem por finalidade de responder questões relacionadas a um objeto de estudo (VIANA, 2014). Sendo assim, é de fundamental importância a realização de estimativas a respeito da avaliação da proposta, quanto ao desenvolvimento de modelos de representação do espaço indoor. Para isso, é importante estabelecer a identificação de alguns objetivos norteadores do processo de testagem, nos quais devem estar alinhados com os objetivos estabelecidos neste trabalho. O processo de avaliação teve como principais objetivos:

OB1: Analisar o esforço realizado na modelagem do ambiente indoor por meio da linguagem de modelagem proposta.

OB2: Avaliar o grau de completude na modelagem dos cenários propostos.

OB3: Analisar a percepção dos usuários quanto as regras de validação fornecidas pela indoorDSL em casos de erros ou não corretude do modelo.

OB4: Analisar o quão fácil foi modelar um ambiente indoor e verificar a qualidade visual da linguagem a fim corrigir possíveis erros oriundos de sobrecarga de informação

Alguns dos aspectos relacionados com a maneira que o usuário percebe o sistema apresentam-se de forma subjetiva e qualitativa. Contudo, é possível complementar a análise com uma estimativa a respeito da eficiência que a ferramenta proporciona, ao medir o esforço requerido para a realização da tarefa por meio do tempo que o usuário leva para a realização das tarefas. Neste trabalho, conforme proposto em (JÚNIOR

et al., 2018), também buscou-se mensurar o tempo que o usuário leva para realizar a tarefa de modelagem proposta, através dos cenários realizados por um expert e os demais participantes. Além disso, buscou-se medir o percentual de corretude dos modelos elaborados por intermédio da ferramenta. Outras abordagens podem ser encontradas na literatura, nas quais o esforço pode ser medido a partir de um marco de referência que pode ser obtido quando se dispõe de uma ferramenta similar, que realize a mesma finalidade, como no caso de (VIANA, 2014; BERTTI, 2015; MARINHO, 2016) ou quando se deseja comparar diferentes versões de uma DSL, como no caso apresentado em (ROSA, 2017).

Dessa forma, pode-se mensurar os resultados e compará-los com o valor tomado como base para realização das tarefas. A medida tida como o valor esperado de execução das tarefas por um usuário expert no domínio é calculada, a partir dos valores de realização das tarefas feitas por dois integrantes que trabalharam junto ao desenvolvimento da infraestrutura INavigS (direta ou indiretamente), e portanto, compreendem bem os conceitos atrelados ao modelo de representação utilizado pela plataforma. Coletados os tempos de realização das tarefas e calculado o valor médio obtido pelos dois integrantes, obtém-se o valor utilizado como base para a realização de cada uma das tarefas propostas. A Tabela 5.1 apresenta os tempos tomados como patamares para a realização das tarefas por especialista no domínio em cada um dos cenários propostos, contendo assim, o valor que será utilizado como referência para aferir com os tempos obtidos pelos participantes nas duas tarefas.

Tabela 5.1 – Tempo gasto utilizado como referência na realização das tarefas propostas.

Tarefa	Descrição	Tempo de realização (mm:ss)
Tarefa fácil	Nesta tarefa embora englobe todos os elementos constituintes do vocabulário da linguagem apresenta uma quantidade de pequena de elementos que precisam constar no modelo.	23:00
Tarefa difícil	Nessa tarefa a quantidade de elementos gráficos utilizados na elaboração das notações gráfica necessárias para se construir o modelo é 280% maior que o modelo proposto na tarefa fácil.	34:00

Uma vez que, definidos os objetivos de avaliação e estimado o esforço para

realização das tarefas propostas, as avaliações visa reunir os dados subjetivos oriundos da percepção que o participante tem a respeito da linguagem indoorDSL, bem como, os dados quantitativos do esforço (tempo) requerido para a concepção dos modelos propostos, nos cenários e a medida de completude (o modelo proposto foi desenvolvido de forma integral ou parcial) com que os modelos são elaborados.

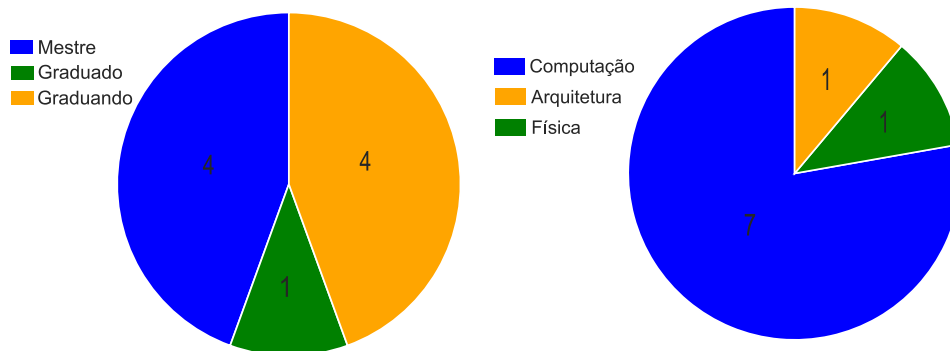
5.1.2 Escolha dos participantes

O processo de escolha dos participantes é um fator determinante no sucesso do processo de avaliação da ferramenta, uma vez que a população presente na amostra pode influenciar diretamente nos resultados obtidos. Uma vez que, na avaliação do esforço requerido para se completar as tarefas pode ser influenciado pela familiaridade com ferramentas de modelagem.

O processo de escolha dos participantes que compõe a amostra utilizada no estudo de avaliação da ferramenta segue o processo de amostragem por conveniência, em vez do modelo tradicional no qual a amostragem é feita de forma aleatória. Conforme descrito por (JÚNIOR et al., 2018), essa abordagem visa a construção de uma população amostral, a partir da disponibilidade apresentada por participantes com pretensão em participar do experimento. Abordagens similares foram seguidas em (González García et al., 2014; JÚNIOR et al., 2018; ROSA, 2017).

Os dados da amostra de participantes coletados a partir do questionário pré-experimento é composta por estudantes de computação (3 participantes), arquitetura (1 participante), analista de tecnologia da informação (1 participante) e alguns professores de informática (3 participantes) e física (1 participante). Sendo que do total, apenas 22,22% (2 participantes) não pertencem a área de computação. O nível de titulação dos usuários é demonstrado através da Figura 5.1a no qual podemos identificar que 4 participantes possuem a titulação de mestre, 1 possui graduação e 4 são graduandos. Quanto os percentuais relacionadas a área de atuação dos indivíduos pode ser visto por meio da Figura 5.1b, onde 7 dos participantes pertencem a computação ou cursos

relacionados a informática, 1 voltado a arquitetura e 1 pertencente a física.



(a) Escolaridade da amostra. (b) Áreas de atuação dos participantes.

Figura 5.1 – Perfil de escolaridade e atuação dos participantes.

Em relação ao nível de conhecimento dos participantes que responderam o formulário prévio de participação do experimento evidenciado pela Figura 5.2, foi possível apurar que 3 participantes afirmaram possuir um conhecimento regular a respeito de DSL, mesma quantidade apresentada em relação a MDE. E que 6 dos participantes já possuíam algum conhecimento sobre ferramentas de modelagem, outro fator importante a salientar é o fato de que apenas 3 dentre os participantes, possuem conhecimento a respeito de modelos de representação do espaço indoor, um dos objetos de estudo deste trabalho.

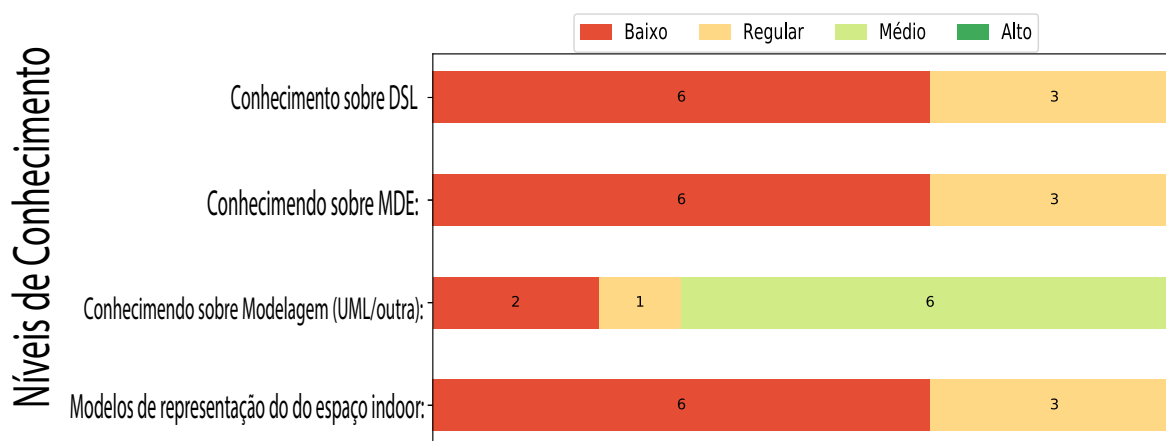


Figura 5.2 – Nível de conhecimento obtido a partir a aplicação do formulário pré-experimento.

Se filtrarmos os dados e se voltarmos somente para os participantes da computação

(ou áreas afins) podemos observar que o conhecimento sobre modelos de representação indoor é “Baixo” por mais da metade dos participantes (4 participantes). Conforme evidenciado pela Figura 5.3.

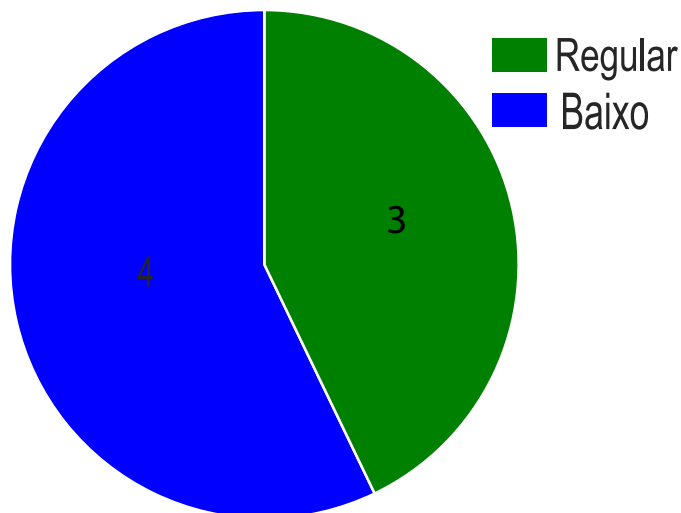


Figura 5.3 – Nível de conhecimento sobre modelo de representação do espaço indoor dentre os participantes da área de computação.

5.1.3 Definição e planejamento da realização dos experimentos

O processo de realização de avaliação, da forma com que o usuário percebe o sistema, foi definido de forma que possa compreender um conjunto de etapas nas quais devem ser seguidas, para que se obtenha êxito no processo de avaliação. Sendo estas abaixo:

Espaço amostral submetido ao experimento: O conjunto de amostras que se mostraram propensas a participar do estudo deve ser composta em sua grande maioria por usuários, que não possuem qualquer conhecimento prévio a respeito de modelagem de ambientes indoor.

Universo de discurso: após as explicações a respeito do domínio e instruções acerca do uso da ferramenta passadas ao usuário, faz-se necessária a elaboração de alguns

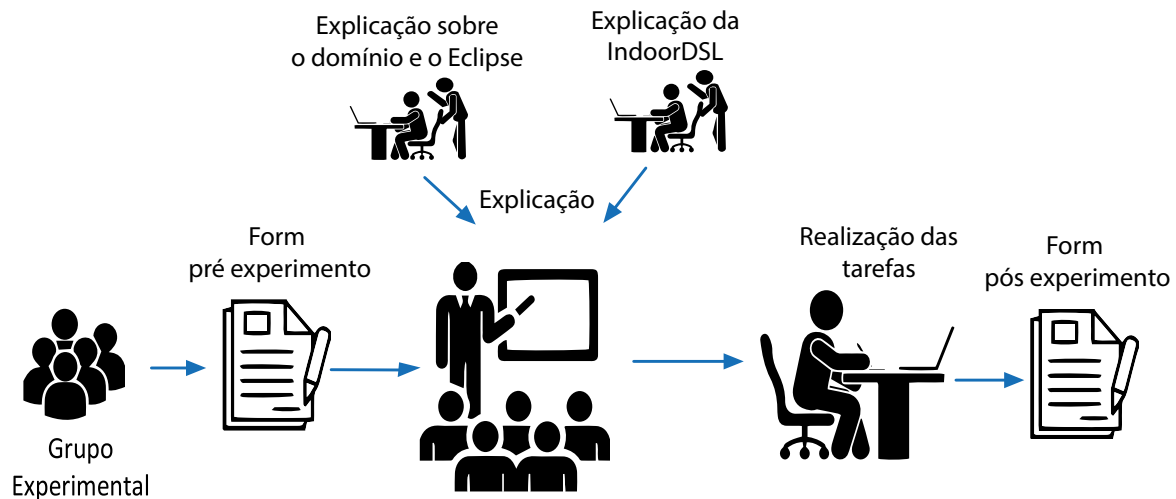


Figura 5.4 – Metodologia de aplicação dos experimentos.

cenários hipotéticos que constituem o uso dos elementos, que fazem parte do domínio em um processo de modelagem, aos quais os usuários seriam submetidos. E assim, possibilitar que os usuários tenham o contato com a linguagem de modelagem através de problemas que se apresentem de forma genérica.

Avaliação: escolhido os participantes e definido os cenários das tarefas a serem realizadas, a sequência natural é apresentar aos usuários os problemas elencados por meio das tarefas, e fornecer o ambiente para que ele possa realizar a modelagem dos cenários propostos. A fim de identificar possíveis erros (sintaxe ou semântico), qualquer outro problema relacionado ao uso da ferramenta e computar as métricas que foram traçadas como objetos de avaliação.

O processo de realização dos experimentos foi conduzido seguindo a abordagem conforme ilustrado na Figura 5.4. Quanto ao fluxo de execução das atividades que norteiam o experimento, é feito por meio de um *check-list* do que é realizado por cada um dos participantes. Conforme exemplificado na Tabela 5.2.

Antes do início da avaliação, cada participante teve que ler e concordar como os termos apresentados nos questionários pré experimento, confirmando (Apêndice A.1) a participação na realização do estudo e que os dados por ele fornecidos serão utilizados

Tabela 5.2 – Tabela com o *checklist* de aplicação do experimento.

Form pré-Experimento	Apresentação	Tarefa Fácil	Tarefa Difícil	Form pós Experimento
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	início:00:00 término:00:00	início:00:00 término:00:00	<input checked="" type="checkbox"/>
		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

apenas para fins de pesquisa, e que será mantida a confidencialidade das informações. Esse primeiro questionário visa coletar informações que dizem respeito a identificação dos usuários e alguns dos conhecimentos prévios a respeito de desenvolvimento com DSL, engenharia de software, programação de computadores, linguagem de modelagem e modelo de representação do espaço indoor. E assim, permitir traçar um perfil dos participantes do experimento.

Uma vez que o questionário pré experimento tenha sido preenchido, o usuário é então submetido a uma explicação a respeito do domínio (compreendendo IPS, INavigS e modelo de representação do espaço indoor), logo após, é apresentado o eclipse como ferramenta de suporte ao ambiente de modelagem e também o conjunto de informações pertinentes a ferramenta IndoorDSL. Somente depois das explicações, o usuário está apto à realização das tarefas. Após finalizar as duas tarefas e ter seu tempo cronometrado, marcando o início e término de cada uma delas, é indicado a responder um questionário pós experimento, contendo as questões que identificam a percepção do usuário em relação ao sistema proposto, a fim de identificar os aspectos relacionados a qualidade da ferramenta.

5.1.4 Procedimento de análise

Para captação do *feedback* do usuário, após o uso da ferramenta, foi aplicado um formulário pós experimento (Apêndice A.2) ao final da modelagem, no qual é constituído de duas seções com propostas distintas. Sendo que a primeira seção do formulário compreende perguntas relacionadas a avaliação da usabilidade por meio do PSSUQ. Tratando-se de um questionário indicado comumente para a realização de coleta de informações do sistema e aplicado amplamente em trabalhos na literatura, como nos




trabalhos em (JÚNIOR et al., 2018; DINIZ, 2016; LEWIS, 2018). O formulário apresenta suas perguntas de forma divididas em 1 avaliação geral e 3 sub avaliações destinadas a avaliar diferentes perspectivas a respeito do sistema a ser estudado(LEWIS, 2018), elas:

- 1 - Visão sobre todo o sistema:** (OVERALL) compreendendo todas as questões de 1 a 16.
- 2 - Utilidade do sistema:** (SYSUSE) compreendendo as questões de 1 a 6.
- 3 - Qualidade da informação:** (INFOQUAL) que compreende as questões de 7 a 12.
- 4 - Qualidade da interface:** (INTERQUAL) compreendendo as questões de 13 a 15.

Além das tradicionais 16 perguntas abrangidas pelo PSSUQ, foram adicionadas mais 3 perguntas buscando identificar a satisfação do usuário quanto ao uso da ferramenta como linguagem de modelagem voltada a criação de modelos de representação do espaço indoor.

A título de avaliação, a pontuação computada ao final permite identificar aspectos direcionados ao objeto, sendo considerada conforme especificado na Tabela 5.3. A forma de pontuação tem como base a escala de Likert por adequada ao permitir que se possa expressar o nível de acordância por meio de declarações que podem ser fortemente positivos (concordo fortemente) e fortemente negativas (discordo fortemente). Sendo que no processo de avaliação utilizado neste trabalho, foram empregados 7 níveis.

Tabela 5.3 – Tabela com as pontuações referentes ao formulário PSSUQ.

Intervalo	Resultado
$Nota \leq 3$	Positivo 
$Nota = 4$	Neutro 
$Nota \geq 5$	Negativo 

A segunda seção contida no formulário engloba as questões relacionadas ao *System Usability Score* (SUS), outro modelo de questionário bastante popular utilizado para avaliação de usabilidade, percebida pelos usuários através de formulários (LEWIS, 2018). A escala de Likert foi novamente aplicada, no entanto, abrangendo apenas 5 níveis.

5.1.5 Ambiente

A realização das tarefas foi realizada de forma remota, sendo que as configurações da máquina na qual os projetos e ambiente estavam hospedados é a seguinte:

- **Sistema Operacional:** Windows 8.1 Single Language 64 bits.
- **Processador:** intel i7 5500U 2.4 GHz.
- **Memória RAM:** 8 GB DDR3.

Este trabalho teve as seguintes variáveis independentes: **1)** O Eclipse Obeo Design Community na versão 11.3, a escolha do Obeo se deu em virtude de fornecer um pacote com todos os plug-ins destinado a metamodelagem já instalados. As ferramentas complementares foram o **2)** Sirius na versão 6.3.0 utilizado para conectar o metamodelo, utilizado para representar o domínio e as ilustrações que fazem parte do vocabulário da linguagem. Além disso, foi utilizado o **3)** Aceleo na versão 6.3.0, que é responsável pela transformação dos modelos, elaborados através da linguagem de modelagem em texto. E por fim, o aplicativo **4)** TeamViewer na versão 15.7.7, no qual foi responsável por estabelecer uma conexão remota máquina-a-máquina, para que pudesse ter acesso ao ambiente de desenvolvimento utilizado na realização das tarefas.

As variáveis dependentes são: **1)** A percepção da qualidade resultante do uso da ferramenta de modelagem, voltada a representação do modelo de representação do espaço indoor. **2)** A completude de realização das tarefas propostas, por meio da linguagem de modelagem desenvolvida. **3)** O esforço requerido para modelar os cenários propostos.

5.2 Tarefas

Cada participante foi incumbido da realização de duas tarefas de modelagem por meio da IndoorDSL, nas quais foram categorizadas como sendo uma simples e uma complexa. Os detalhes acerca dos cenários propostos nas tarefas são:

Fácil

Sendo a tarefa inicial a ser desenvolvida englobando um número pequeno de elementos, nos quais o modelo deve conter. Todas as informações de propriedades dos elementos contidos no modelo foram fornecidas, para que o usuário pudesse desempenhar a sua tarefa. A tarefa consistia em criar um modelo de representação do espaço indoor, a partir de um modelo IFC como referência contendo um cenário hipotético no qual apresenta a organização do ambiente. O referido cenário pode ser visto por meio da Figura 5.5.

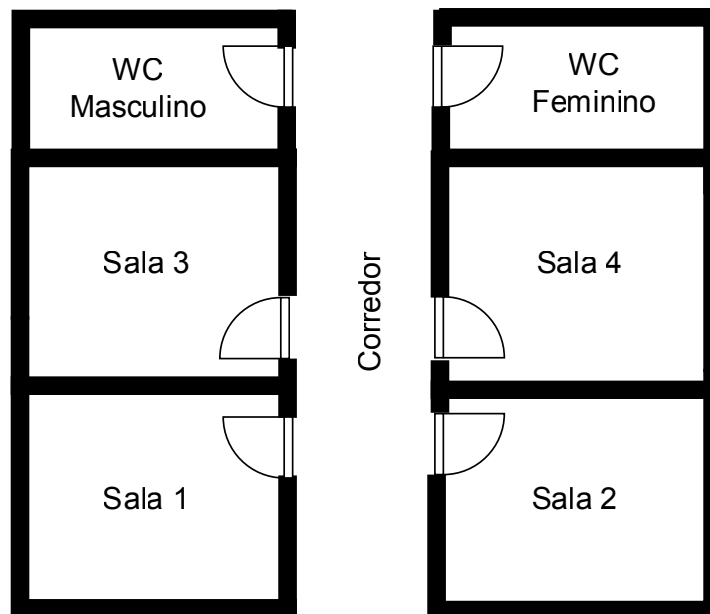


Figura 5.5 – Modelo de referência proposto para a primeira tarefa.

No cenário apresenta 6 salas distintas e um corredor, totalizando 7 elementos arquiteturais (*BuildElement*), além disso, foi assumido que cada elemento que compõe o prédio receberia um beacon associado a ele. Todas as informações necessárias para essa tarefa foram informadas por meio de um documento de texto. A Tabela 5.4 apresenta os dados direcionados ao preenchimento dos atributos contidos no objeto *Address*, enquanto que na Tabela 5.5 contém os dados referentes as propriedades do objeto *Build* no qual necessita somente do valor referente ao atributo “name”.

Em relação as informações relacionadas aos elementos do prédio (*BuildElements*),

Tabela 5.4 – Dados para Address - Tarefa fácil.

Dados de Endereço	
City	São Luís
State	Maranhão
Street	Avenida Edson Brandão
Zip Code	65045-380
Complement	sem complemento
Number	01
Neighborhood	Anil

Tabela 5.5 – Informações do elemento Build - Tarefa fácil.

Dados do Prédio	
Nome	Prédio Comercial São Luís

todos os dados necessários para preencher as propriedades estão explícitos na imagem de representação do cenário. Nos quais são descritos pela Tabela 5.6 os valores para o nome e descrição, foram assumidos como sendo os mesmos a título de simplificar a tarefa.

Tabela 5.6 – Informações relacionadas ao BuildElements - Tarefa fácil.

Informações referentes a cada BuildElements					
Local	Name	Description	Type	Accessible	Vertical
WC Masculino	WC Masculino	WC Masculino	ROOM	true	false
WC Feminino	WC Feminino	WC Feminino	ROOM	true	false
Sala 1	Sala 1	Sala 1	ROOM	true	false
Sala 2	Sala 2	Sala 2	ROOM	true	false
Sala 3	Sala 3	Sala 3	ROOM	true	false
Sala 4	Sala 4	Sala 4	ROOM	true	false
Corredor	Corredor	Corredor	PATH	true	false

Os dados relacionados aos atributos dos beacons para cada uma das posições em relação aos respectivos ambientes pode ser vista na Tabela 5.7:

Tabela 5.7 – Informações relacionadas aos *Beacons* - Tarefa fácil.

Informações referentes aos Beacons			
Local	Beacon/Latitude	Beacon/Longitude	Beacon/Código
WC Masculino	-2.556346	-44.242318	WCM1
WC Feminino	-2.556412	-44.242262	WCF1
Sala 1	-2.556292	-44.242258	S01
Sala 2	-2.556356	-44.242199	S02
Sala 3	-2.556328	-44.242289	S03
Sala 4	-2.556390	-44.242235	S04
Corredor	-2.556352	-44.242252	C01

Quanto ao elemento `Link` o único valor que é necessário fornecer, trata-se do campo “weight”, que para simplificação do modelo foi assumindo sendo o valor 4 para todas as suas ocorrências. Já as ligações entre beacons são definidas segundo os dados apresentados na Tabela 5.8.

Tabela 5.8 – Informações relacionadas ao `Link`.

Informações referentes aos Links - Tarefa 1		
Origem	Destino	Weight
Sala 1	Corredor	4
Sala 2	Corredor	4
Sala 3	Corredor	4
Sala 4	Corredor	4
WC Masculino	Corredor	4
WC Feminino	Corredor	4

5.2.1 Difícil

A fim de adicionar mais complexidade no processo de desenvolvimento, foi proposto a realização de uma tarefa que apresentasse uma quantidade maior de elementos no modelo. Isto poderia dificultar seu desenvolvimento em virtude de uma melhor organização dos elementos na área de modelagem, e a quantidade de relações e dados necessários para sua efetiva conclusão. Sendo assim, o cenário hipotético segue abordagem similar à tarefa fácil, conforme o mapa indicando qual modelo se deve desenvolver ilustrado por meio da Figura 5.6.

O modelo de referência apresenta um total de 14 `BuildElements`, sendo 13 salas e um corredor. As informações relacionadas aos atributos referentes ao objeto `Address` podem ser vistas na Tabela 5.9 nos quais são reflexos do que foi extraído na fase de análise do domínio. Já os dados necessários para o preenchimento das informações relacionadas a `Build`, podem ser visto por meio da Tabela 5.10, no qual apresenta somente o atributo "nome".

As informações relacionadas aos dados atribuídos aos beacons ao serem relacionados ao respectivo elemento do prédio são expressos na Tabela 5.11. Contendo a

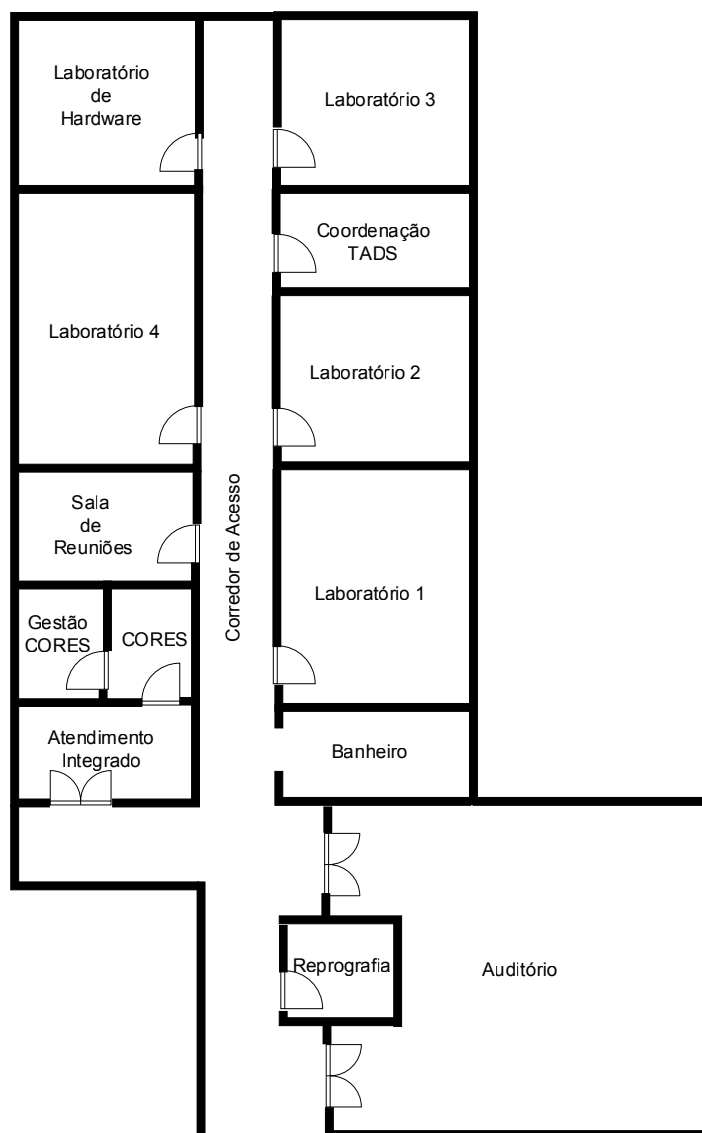


Figura 5.6 – Modelo de referência proposto para a segunda tarefa.

identificação do respectivo elemento do prédio, no qual o beacon deve ser colocado, sua posição geográfica (latitude e longitude) e o campo código.

Em se tratando dos Links, as informações necessárias para preenchimento do modelo são informadas por meio da Tabela 5.12. Com os dados de origem e destino e o campo “Weight”. O modelo gerado por meio da linguagem de modelagem pode ser visto no Apêndice B, bem como o código resultante da sua transformação (Apêndice C).

Tabela 5.9 – Dados para Address-Tarefa 2.

Dados de Endereço	
City	Araguaína
State	Tocantins
Street	Avenida Paraguai esquina com Avenida Amazonas, Quadra 56
Zip Code	77824-838
Complement	Prédio anexo I
Number	01
Neighborhood	Cimba

Tabela 5.10 – Informações do elemento Build

Dados do Prédio	
Nome	Prédio administrativo e laboratórios da ala Oeste

Tabela 5.11 – Informações relacionadas ao *Beacons* - Tarefa 2.

Informações referentes aos Beacons			
Local	Beacon/Latitude	Beacon/Longitude	Beacon/Código
Auditório	-7.181542	-48.193646	BN001
Reprografia	-7.181569	-48.193720	SN001
Banheiro	-7.181569	-48.193708	SN004
Atendimento integrado	-7.181484	-48.193772	CN003
CORES	-7.181463	-48.193746	C1S001
Gestão Cores	-7.181462	-48.193781	RS001
Laboratório 1	-7.181469	-48.193695	SS004
Laboratório 2	-7.181395	-48.193685	HN003
Laboratório 3	-7.181304	-48.193678	HS002
Laboratório 4	-7.181359	-48.193765	HN004
Coordenação TADS	-7.181356	-48.193681	HS004
Laboratório de Hardware	-7.181305	-48.193759	HS005
Sala de Reuniões	-7.181405	-48.193764	HS006
Corredor de Acesso	-7.181429	-48.193724	HS007

5.3 Resultados

Realizada as tarefas e obtidos os tempos de realização individual para cada participante, pode-se estimar a mediana para cada um dos cenários propostos, conforme ilustrado pela Figura 5.7, que apresenta o boxplot contendo as informações relacionadas ao tempo de execução das tarefas. Nesse tipo de gráfico, é evidenciado a distribuição dos valores obtidos no experimento através de caixas, no qual a linha contida dentro da caixa representa a mediana dos valores obtidos para cada cenário. Enquanto as duas

Tabela 5.12 – Informações relacionadas ao Link .-Tarefa Difícil

Informações referentes aos Links - Tarefa 2		
Origem	Destino	Weight
Auditório	Corredor	4
Reprografia	Corredor	4
Banheiro	Corredor	4
Atendimento Integrado	Cores	4
Atendimento Integrado	Corredor	4
Cores	Gestão Cores	4
Laboratório 1	Corredor	4
Laboratório 2	Corredor	4
Laboratório 3	Corredor	4
Laboratório 4	Corredor	4
Sala de Reuniões	Corredor	4
Coordenação TADS	Corredor	4
Laboratório de Hardware	Corredor	4

delimitações na forma de “T” abaixo e acima da caixa indicam o menor valor (a tarefa realizada mais rápida) e o maior (a tarefa mais demorada), obtidos respectivamente dentre todos os valores válidos. Tendo em vista a realização das tarefas e a cronometragem do tempo, pode-se observar pelo gráfico que não foi obtido nenhum valor que pudesse representar um *outlier* dentre os valores. Ao avaliar a dispersão dos resultados, podemos observar que a tarefa difícil (cenário 2) apresentou uma maior variabilidade entre os seus resultados o que evidencia que a tarefa fácil (cenário 1) se obteve um melhor resultado em relação aos tempos obtidos.

Uma visão mais abrangente dos tempos, pode ser observada na Tabela 5.14, na qual apresenta os tempos gastos por todos os participantes em minutos nos dois cenários propostos. Em que na quarta coluna, podemos observar a variabilidade percentual em relação ao tempo estimado, para realização das tarefas e por último na quinta coluna é encontrado os valores relacionados ao acerto na modelagem que foi obtido, a partir da análise dos modelos após finalizada a tarefa de cada participante. Para que se pudesse definir e categorizar o grau de acertos, fez-se necessário estabelecer um valor quantificável por cada categoria, seguindo as especificações da Tabela 5.13, sendo que 100% de acertos significa que o modelo proposto foi modelado em sua totalidade pelo usuário.

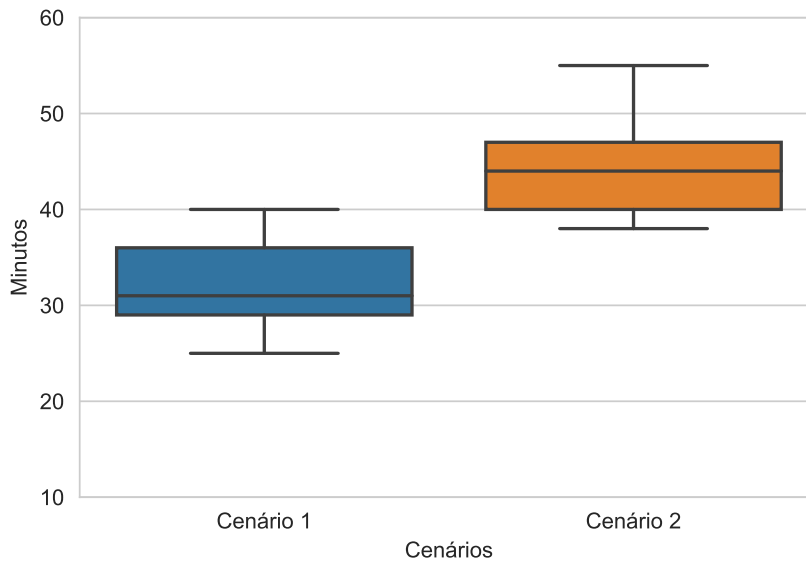


Figura 5.7 – Boxplot criado contendo os dados do tempo de duração das tarefas.

Tabela 5.13

SIM	PARCIAL	NÃO
Acertos=100%	$75\% \leq \text{acertos} < 100$	$\text{acertos} < 75\%$

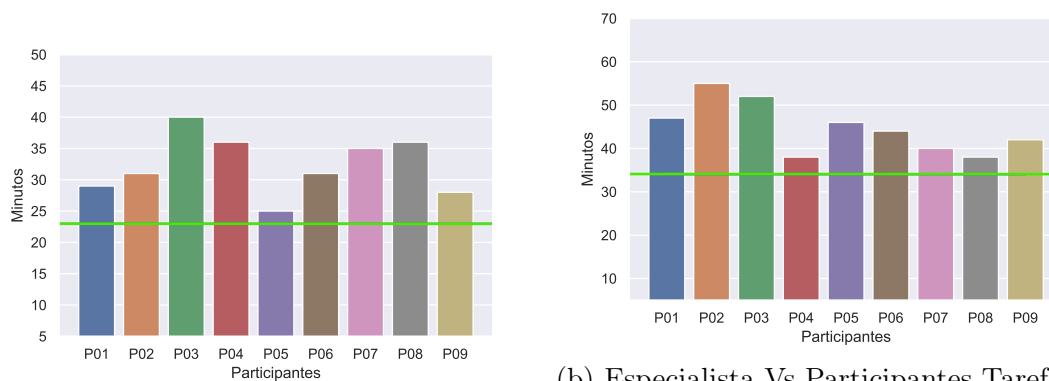
A forma de computar os valores obtidos é realizada por meio das etapas descritas por Padrini-Andrade et al. (2019), no qual é possível encontrar uma tabela contendo a tradução das questões.

A partir da análise dos tempos computados podemos perceber que o esforço exigido pela realização da tarefa fácil apresenta percentuais mais altos quando relacionado a tarefa difícil em virtude do primeiro contato com a ferramenta e o processo de familiarização com a linguagem. Como evidenciado nos tempos obtidos no segundo experimento, 55,55% dos participantes (P03, P04, P06, P07 e P08) tiveram seu esforço reduzido por haver uma melhor compreensão sobre o uso da ferramenta, mesmo apesar da tarefa difícil conter mais que o dobro de componentes necessários para sua elaboração, quando comparada a tarefa fácil. Um comparativo entre o valor de referência e o tempo obtido pelos participantes, pode ser visto por meio da Figura 5.8a para a tarefa fácil e pela Figura 5.8b para a

Tabela 5.14 – Tempo gasto por participante em relação aos experimentos.

Cenário	Participante	Tempo em minutos	Valor de referência $\pm\Delta$ %	Modelagem correta
Tarefa fácil Tempo Estimado: (00:23)	P01	29	+26,09%	SIM
	P02	31	+34,78%	SIM
	P03	40	+73,91%	PARCIAL
	P04	36	+56,52%	SIM
	P05	25	+8,70%	SIM
	P06	31	+34,78%	SIM
	P07	35	+52,17%	SIM
	P08	36	+56,52%	SIM
	P09	28	+21,74%	PARCIAL
Tarefa difícil Tempo Estimado (00:34)	P01	47	+38,24%	SIM
	P02	55	+61,76%	PARCIAL
	P03	52	+52,94%	PARCIAL
	P04	38	+11,76%	SIM
	P05	46	+35,29%	PARCIAL
	P06	44	+29,41%	SIM
	P07	40	+17,65%	SIM
	P08	38	+11,76%	SIM
	P09	42	+23,53%	PARCIAL

tarefa difícil. Onde a linha verde representa o valor de referência obtido pelo especialista no domínio.



(a) Especialista Vs Participantes Tarefa fácil. cil.

(b) Especialista Vs Participantes Tarefa difícil.

Figura 5.8 – Relação entre os tempos obtidos pelos participantes x especialista.

Os resultados obtidos visam atender os objetivos de avaliação traçados inicialmente no momento de especificação dos itens aos quais os usuários avaliarão. Portanto, dada a ordem dos objetivos. **OB1** (analisar o esforço), que tem por finalidade avaliar o esforço requerido para modelagem por meio da IndoorDSL. Enquanto que **OB2** (avaliar se a tarefa foi feita em sua totalidade), diz respeito a avaliação da completude dos modelos

elaborados pelos participantes é demonstrada pelos gráficos ilustrados pelas Figura 5.9a e Figura 5.9b para a tarefa fácil e tarefa difícil, respectivamente. Pelo qual demonstra que mesmo havendo um maior esforço para a realização da tarefa fácil, ainda assim foi obtido um maior êxito na sua realização, indicando que um total 7 participantes construíram seus modelos, sem a ausência de qualquer elemento. Enquanto que, no cenário difícil apenas 5 participantes dos usuários completaram a tarefa em sua totalidade.

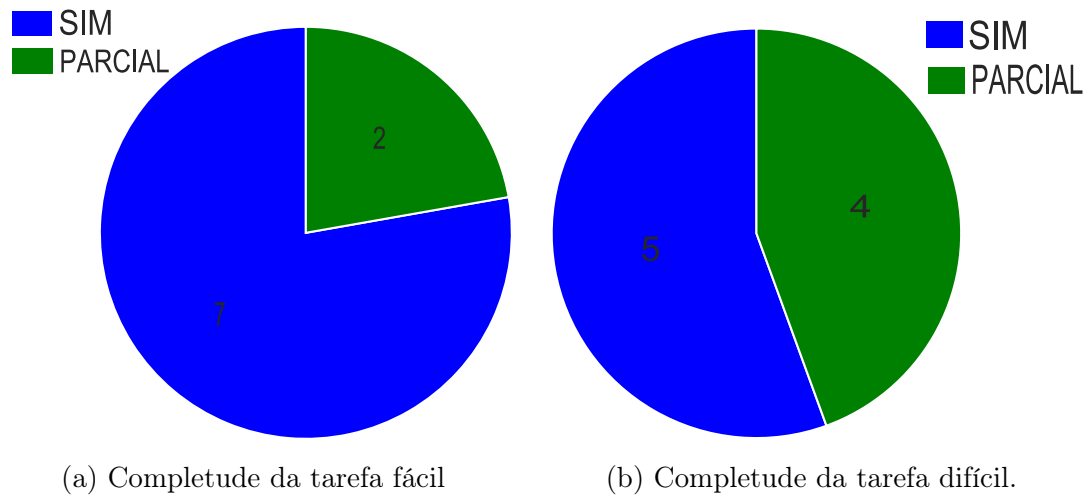


Figura 5.9 – Quantidade de participantes X Tarefas resolvidas em sua totalidade.

No que diz respeito aos indicadores responsáveis por avaliar os objetivos **OB3** (análise da percepção do usuário quanto as regras de validação) e **OB4** (análise da qualidade visual da ferramenta como linguagem de modelagem), são obtidos pelo resultado coletados a partir das questões apresentadas no questionário PSSUQ, que tem por finalidade identificar se as regras de validação especificadas na Seção 4.3, por meio das expressões especificadas no momento de concepção da linguagem visual, permitem tratar erros e garantir a corretude do modelo ao evitar que haja ambiguidades. O resultado do questionado PSSUQ pode ser visto por meio da Figura 5.10, demonstrando que não houve pontos negativos na categoria que engloba os aspectos relacionados ao **OB3** INFOQUAL. Enquanto que na categoria correspondente ao **OB4** INTERQUAL apresentou 96% de pontuação positiva. Já as categorias restantes (OVERALL e SYSUSER) mantiveram os patamares acima 98%.

Ainda em relação ao feedback do usuário, foi possível traçar uma outra perspectiva

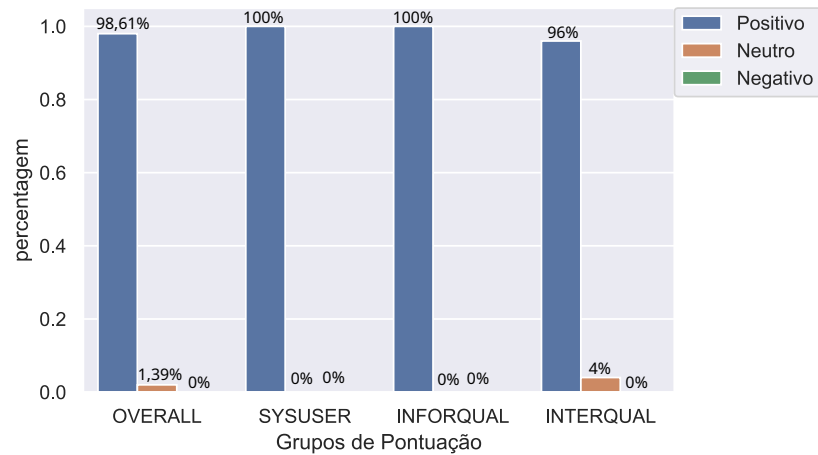


Figura 5.10 – Resultado PSSUQ por categorias.

por meio da aplicação do formulário SUS, na segunda seção do formulário pós experimento (Ver apêndice A.2 Figura A.10) no qual o valor obtido foi de 87,22 pontos conforme ilustrado na Figura 5.11 considerado um valor excelente de acordo com a escala.

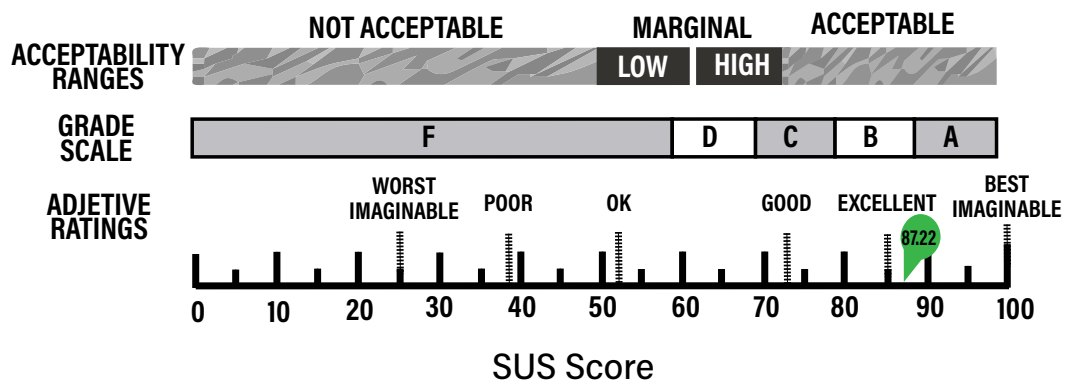


Figura 5.11 – Resultado obtido pelo SUS score.

5.4 Ameaças à validade

Em relação as ameaças identificadas durante o processo de realização desse estudo foram identificadas como sendo:

Quantidade de participantes submetidos ao teste: A quantidade de indivíduos que se mostraram propensos a realização do teste foi baixa, devido a fatores

externos, causados por uma crise sanitária e de saúde pública. Diante disso, os testes não puderam ser realizados em laboratórios de informática, com participantes de diversas áreas científicas e em um número maior de amostras.

Quantidade de testes: Em virtude do tempo demandado pela realização das tarefas, pode ter sido ampliado devido à realização ter ocorrido de forma remota. Apenas dois cenários foram propostos, a fim de evitar a fadiga dos participantes, fator no qual pode ocasionar a desistência na participação do experimento. Mesmo sendo apenas dois testes, o tempo necessário para a realização das tarefas acrescidos dos dois formulários que deveriam ser preenchidos, pode ter sido um fator inibidor, o que explicaria a baixa adesão de participantes.

Realização dos testes de forma remota: A realização dos testes de forma remota foi um fator impactante na determinação do tempo, medida utilizada para mensurar o esforço necessário para a modelagem dos cenários propostos, nas tarefas por meio da linguagem indoorDSL. Uma vez que, pode sofrer influência direta pelas especificações de hardware da máquina utilizada pelo participante (máquina remota) e pela degradação da largura de banda utilizada na comunicação ou ainda, o tipo de rede a qual o usuário esteja conectado no momento do teste.

Com o intuito de mitigar as ameaças que poderiam influenciar no processo de realização das tarefas e na avaliação do trabalho, por parte dos participantes envolvidos, foram tomadas as seguintes medidas:

Iniciativa 1

A adoção do SUS auxiliou a medir as características subjetivas inerentes a qualidade percebida por parte do usuário uma vez que seu resultado não sofre alterações bruscas em virtude do aumento do espaço de amostras. Embora o espaço amostral de participantes seja constituído por uma quantidade pequena de indivíduos, ainda assim, é possível mensurar a percepção dos usuários sob os aspectos direcionados ao uso da

ferramenta proposta neste trabalho. Neste sentido, segundo Turner et al. (2006), é possível descobrir problemas mesmo com a utilização de um único teste, apresentando algumas importantes reivindicações quanto a realização do experimento em relação ao número de participantes, sendo elas:

- A maioria dos problemas de usabilidade são identificados pelas primeiras três a cinco pessoas.
- Executar testes adicionais improvavelmente resultará em novas informações.
- Maximização do retorno sobre o investimento, ao submeter os testes a um pequeno grupo usando uma metodologia interativa.

Iniciativa 2

Apesar da quantidade de testes ter se mostrado pequena, a modelagem das tarefas propostas, por meio da linguagem de modelagem IndooDSL, englobam todo o vocabulário e notações gráficas possíveis fornecidas pela linguagem. O aumento da quantidade de tarefas poderia resultar em desistências ou ser uma condição determinante para que os usuários não se submetessem aos testes. Sendo assim, buscou-se elaborar as tarefas de modo que esforço requerido para a modelagem não se tornasse um fator que pudesse afugentar os participantes. Passando a ser um fator determinante durante a sua concepção.

Iniciativa 3

O esforço de realização da modelagem das tarefas propostas sendo um dos indicadores utilizados como medida neste trabalho, foi fortemente influenciado pela realização remota. No entanto, as métricas de referência utilizadas para título de comparação com os participantes foram obtidas através da reprodução de forma mais fiel possível do ambiente, em que outros participantes seriam submetidos. Além da confiabilidade, uma vez que os participantes não sofreram interferência ao longo de realização do seu processo

de modelagem das tarefas. Nas ocasiões em que ocorreram queda de internet, a tarefa foi paralisada, e retomada do ponto de parada, sem que ocorresse acréscimo ao tempo total.

5.5 Considerações Finais

Esse capítulo teve a finalidade de avaliar a linguagem de modelagem IndoorDSL por meio de experimentos nos quais participantes realizam tarefas por meio da linguagem, e ao responder alguns formulários, pode-se captar a percepção dos usuários em relação a linguagem. Tais experimentos foram conduzidos e planejados visando traçar estimativas mensuráveis, e a partir de tarefas, questionários e coleta de dados, pode-se fornecer os resultados das estatísticas e assim, atestar o quão a linguagem possui de qualidade.

Os experimentos permitiram verificar a eficácia da DSL no processo de modelagem dos modelos de representação do espaço indoor utilizados pela INavigS. Além disso, o *feedback* positivo dos usuários permitiu verificar o quão promissora a abordagem do uso de modelos pode ser benéfica quando aplicada nesse contexto. A realização dos testes serviram para captar os aspectos subjetivos relacionados a qualidade fornecida pela linguagem, que nos dois testes PSSUQ e SUS apresentaram resultados satisfatórios e assim, atestando a sua qualidade e atendendo os objetivos traçados inicialmente no planejamento dos experimentos.

6 Conclusão, Publicações e Trabalhos Futuros

Neste capítulo é apresentado as principais contribuições deste projeto de dissertação, no qual foi desenvolvido uma DSL para modelagem de ambientes indoor, com base no modelo de representação do espaço utilizado pela infraestrutura de software INavigS. Onde são listadas as principais contribuições alcançadas com o trabalho e enumerados algumas das possíveis contribuições oferecidas pela linguagem desenvolvida.

6.1 Contribuições do trabalho

Contribuições do projeto apresentado neste trabalho de mestrado:

- Realização da análise do domínio (modelos de representação do espaço indoor) e concepção do metamodelo, pelo qual se permite especificar uma linguagem de modelagem, com conceitos oriundos do próprio domínio.
- Construção de uma DSL que permita elaborar modelos de representação do espaço indoor (IndoorDSL), fazendo com que a realização da tarefa de criação do modelo se torne mais intuitiva, por especificar modelos por meio de notações gráficas, respeitando as especificações estabelecidas na definição de sua sintaxe concreta.
- Fornecer uma linguagem de modelagem dotada de um conjunto de regras de validação, nos quais permite os usuários criarem seus próprios modelos e verificarem a presença de inconsistências (ausência de elementos, valores errados para um dado atributo e etc.) no modelo.
- Permitir que os modelos gerados por meio da linguagem de modelagem possam ser transformados em artefatos de código (transformação M2T). E assim, torna-os manipuláveis pelo INavigS.

- Realização da análise referente as avaliações experimentais nas quais a DSL foi submetida.

6.2 Limitações da proposta

A seguir são, apresentadas algumas das limitações identificadas na linguagem de modelagem desenvolvida neste projeto de dissertação, sendo elas:

- Embora a DSL forneça um meio de construção dos modelos de representação do espaço indoor em um elevado nível de abstração por meio de uma interface gráfica, ainda assim, a linguagem tem a modelagem de sua sintaxe concreta limitada a ferramenta utilizada para sua especificação. Isso faz com que seja necessário o conhecimento acerca da ferramenta e configuração do ambiente, antes de se iniciar o processo de modelagem. Fazendo com que o Eclipse se torne um pré-requisito para o uso da ferramenta.
- A ferramenta procura solucionar uma lacuna apresentada pelo INavigS, o modelo gerado é voltado a essa estrutura de software como IPS. Não identificando outros IPS que possam vir a usufruir dos benefícios advindos do uso da linguagem desenvolvida neste trabalho.
- O sucesso no processo de modelagem está atrelado ao conhecimento prévio a respeito do domínio, o sucesso e completude dos modelos dependem inteiramente do quão ciente o usuário estar sobre as regras de negócio compreendidas pelo domínio.
- A transformação do modelo gera artefatos de código somente na linguagem Java.

6.3 Publicações

A fim de expor os resultados obtidos, a partir da pesquisa que norteou essa dissertação de mestrado, foi realizada a publicação de um artigo científico em periódico

conforme descrito a seguir:

- ANDRADE, E. D. S.; VALE, S. B. Indoor-dsl: A model driven approach to modeling indoor representation models for a navigation infrastructure. *International Journal of Computer Applications*, Foundation of Computer Science (FCS), NY, USA, New York, USA, v. 176, n. 42, p. 7–14, Jul 2020. ISSN 0975-8887. Disponível em: <<http://www.ijcaonline.org/archives/volume176/number42/31481-2020920538>>. Publicado em Julho de 2020.

Qualis CAPES (2019): A4 em Ciência da Computação.

Situação: Publicado.

6.4 Trabalhos Futuros

Embora a linguagem de modelagem desenvolvida forneça suporte a elaboração de modelos de representação do espaço indoor, ainda assim foi possível identificar algumas melhorias que podem ser realizadas. Tais como:

1. Criação de um manual contendo a descrição da linguagem, discriminação dos elementos e procedimentos para o processo de criação das notações;
2. Desenvolver futuras versões da linguagem englobando outros modelos de representação do espaço indoor e modelos de base para outros IPS;
3. Criar um conjunto de regras que permita o modelo ser transformado em mesmo nível de abstração (PIM para PIM);
4. Permitir que o modelo possa ser transformado para outros modelos em níveis de abstração subjacentes (PIM para PSM);

5. Realização de testes de usabilidade englobando um número maior de pessoas com perfis mais diversificados. A fim de identificar melhorias e fornecer uma análise mais aprofundada.

Referências Bibliográficas

- ARRUDA, D. **QualiBD Tool: Implementation Details**. Cornell University, 2019. Eprint: 1912.03866. Disponível em: <<https://arxiv.org/abs/1912.03866?gathStatIcon=true>>. Citado 2 vezes nas páginas 47 e 67.
- BENOUDA, H.; AZIZI, M.; ESBAI, R.; MOUSSAOUI, M. Code generation approach for mobile application using acceleo. **International Review on Computers and Software (IRECOS)**, v. 11, p. 160, 02 2016. ISSN 2533-1728. Disponível em: <[https://www.praiseworthyprize.org/jsm/index.php?journal=irecos&page=article&op=view&path\[\]=18580](https://www.praiseworthyprize.org/jsm/index.php?journal=irecos&page=article&op=view&path[]=18580)>. Citado na página 22.
- BERTTI, E. **MIRA: Um Ambiente Para Interfaces Dirigidas Por Modelos Para Aplicações Rest**. 138 p. Dissertação (Mestrado) — UPONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO - PUC-RIO, mar 2015. Departamento de Informática. Disponível em: <<https://www.maxwell.vrac.puc-rio.br/colecao.php?strSecao=resultado&nrSeq=25307@1>>. Citado na página 82.
- BORELLI, H. **Uma linguagem de modelagem de domínio específico para linhas de produto de software dinâmicas**. Dissertação (Mestrado) — Universidade Federal de Goiás, 2016. Instituto de Informática - INF (RG). Disponível em: <<http://repositorio.bc.ufg.br/tede/handle/tede/5893>>. Citado 2 vezes nas páginas 36 e 45.
- BRADLEY, A.; LI, H.; LARK, R.; DUNN, S. Bim for infrastructure: An overall review and constructor perspective. **Automation in Construction**, v. 71, p. 139–152, 11 2016. Citado na página 29.
- BRAMBILLA, M.; CABOT, J.; WIMMER, M. **Model-Driven Software Engineering in Practice**. 1nd. ed. [S.l.]: Morgan & Claypool Publishers, 2012. v. 1. 1–182 p. Synthesis Lectures on Software Engineering. ISSN 2328-3319. ISBN 9781627057080. Citado 3 vezes nas páginas 11, 36 e 38.
- BRAMBILLA, M.; CABOT, J.; WIMMER, M. **Model-Driven Software Engineering in Practice: Second Edition**. 2nd. ed. [S.l.]: Morgan & Claypool Publishers, 2017. 1-207 p. Synthesis Lectures on Software Engineering. ISBN 1627057080. Citado 3 vezes nas páginas 11, 35 e 37.
- CHALHOUB, N.; SIMON-NAGY, G. Indoor navigation based on linked data at honvéd hospital, budapest. In: **2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI)**. IEEE, 2018. Disponível em: <<https://doi.org/10.1109/saci.2018.8440964>>. Citado na página 50.

- Chan, S.; Wu, P.; Fu, L. Robust 2d indoor localization through laser slam and visual slam fusion. In: **2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)**. [S.l.]: IEE, 2018. p. 1263–1268. ISBN 978-1-5386-6650-0. Conference Location: Miyazaki, Japan, Japan. Citado 3 vezes nas páginas 28, 57 e 60.
- CHEN, J.; CLARKE, K. Indoor cartography. **Cartography and Geographic Information Science**, p. 1–15, 06 2019. Citado 2 vezes nas páginas 29 e 31.
- CHENG, B. H.; COMBEMALE, B.; FRANCE, R. B.; JÉZÉQUEL, J.-M.; RUMPE, B. Globalizing domain-specific languages (dagstuhl seminar 14412). In: SCHLOSS DAGSTUHL-LEIBNIZ-ZENTRUM FUER INFORMATIK. **Dagstuhl Reports**. [S.l.], 2015. v. 4, n. 10. Citado na página 32.
- COMBEMALE, B. **Metamodeling Approach for Model Simulation and Verification Application to Process Engineering**. Tese (Theses) — Institut National Polytechnique de Toulouse - INPT, jul. 2008. Disponível em: <<https://tel.archives-ouvertes.fr/tel-00321863>>. Citado 2 vezes nas páginas 11 e 42.
- DEURSEN, A. V.; KLINT, P.; VISSER, J. Domain-specific languages: An annotated bibliography. **ACM Sigplan Notices**, ACM, v. 35, n. 6, p. 26–36, 2000. Citado 2 vezes nas páginas 38 e 61.
- DINIZ, H. B. M. **Linguagem específica de domínio para abstração de solução de processamento de eventos complexo**. 112 p. Dissertação (Mestrado) — Universidade Federal de Pernambuco, mar 2016. Centro de Informática. Disponível em: <<https://repositorio.ufpe.br/handle/123456789/18030>>. Citado 2 vezes nas páginas 80 e 88.
- ECLIPSEORG. **Specifying Diagram Editors**. 2020. [Online; acessado em 14-Julho-2020]. Disponível em: <https://www.eclipse.org/sirius/doc/specifier/diagrams/Diagrams.html#layers_graphical_elements>. Citado na página 68.
- ECLIPSEORG. **Specifying Viewpoints**. 2020. [Online; acessado em 14 de Julho de 2020]. Disponível em: <https://www.eclipse.org/sirius/doc/specifier/general/Specifying_Viewpoints.html>. Citado na página 46.
- ECLIPSEORG. **What is Acceleo?** 2020. [Online; acessado em 14 de Junho de 2020]. Disponível em: <<https://www.eclipse.org/acceleo/overview.html>>. Citado na página 47.
- EINARSSON, A. F.; PATREKSSON, P.; HAMDQA, M.; HAMOU-LHADJ, A. SmartHomeML: Towards a domain-specific modeling language for creating smart home applications. In: **2017 IEEE International Congress on Internet of Things (ICIOT)**. IEEE, 2017. p. 82–88. Disponível em: <<https://doi.org/10.1109/ieee.iciot.2017.35>>. Citado na página 33.

- Endo, Y.; Sato, K.; Yamashita, A.; Matsubayashi, K. Indoor positioning and obstacle detection for visually impaired navigation system based on lsd-slam. In: **2017 International Conference on Biometrics and Kansei Engineering (ICBAKE)**. [S.l.]: IEEE, 2017. p. 158–162. ISBN 978-1-5386-3401-1. Conference Location: Kyoto, Japan. Citado 3 vezes nas páginas 28, 57 e 60.
- FADLI, F.; KUTTY, N.; WANG, Z.; ZLATANOVA, S.; MAHDJOUBI, L.; BOGUSLAWSKI, P.; ZVEROVICH, V. Extending indoor open street mapping environments to navigable 3d citygml building models: Emergency response assessment. **ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, XLII-4, p. 161–168, 09 2018. Citado 2 vezes nas páginas 57 e 60.
- FOWLER, M. **Domain Specific Languages**. 1st. ed. [S.l.]: Addison-Wesley Professional, 2010. ISBN 0321712943. Citado na página 38.
- GAO, R.; YE, F.; LUO, G.; CONG, J. Indoor map construction via mobile crowdsensing. In: _____. [S.l.: s.n.], 2018. p. 3–30. ISBN 978-981-10-8377-8. SpringerBriefs in Computer Science. Citado 2 vezes nas páginas 57 e 60.
- GARCÍA, C. G.; GARCÍA-BUSTELO, B. P.; ESPADA, J.; CUEVA-FERNANDEZ, G. Midgar: Generation of heterogeneous objects interconnecting applications. a domain specific language proposal for internet of things scenarios. **Computer Networks**, v. 64, 05 2014. Citado na página 22.
- GASEVIC, D.; DJURIC, D.; DEVEDZIC, V. **Model Driven Engineering and Ontology Development**. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2009. ISBN 3642002811. Citado na página 44.
- González García, C.; Pelayo G-Bustelo, B. C.; Pascual Espada, J.; CUEVA-FERNANDEZ, G. Midgar: Generation of heterogeneous objects interconnecting applications. a domain specific language proposal for internet of things scenarios. **Computer Networks**, v. 64, p. 143 – 158, 2014. ISSN 1389-1286. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1389128614000528>>. Citado na página 83.
- GRAICHEN, T.; QUINGER, S.; HEINKEL, U.; STRASSENBURG-KLECIK, M. A novel, user-friendly indoor mapping approach for openstreetmap. In: . Elektrotbit Automotive, 2017. v. 12. Universidade de Tecnologia de Chemnitz, Chemnitz. Disponível em: <<https://nbn-resolving.org/urn:nbn:de:bsz:ch1-qucosa-221082>>. Citado 2 vezes nas páginas 57 e 60.
- HAMMOUDI, S.; PIRES, L. F.; SELIC, B. (Ed.). **Model-Driven Engineering and Software Development**. Springer International Publishing, 2019. Disponível em: <<https://doi.org/10.1007/978-3-030-11030-7>>. Citado na página 34.

- HU, X.; FAN, H.; ZIPF, A.; SHANG, J.; GU, F. A conceptual framework for indoor mapping by using grammars. In: . [S.l.: s.n.], 2017. IV-2/W4, p. 335–342. Citado 2 vezes nas páginas 58 e 60.
- HUANG, C.-H.; LEE, L.-H.; HO, C. C.; WU, L.-L.; LAI, Z.-H. Real-time rfid indoor positioning system based on kalman-filter drift removal and heron-bilateration location estimation. **IEEE Transactions on Instrumentation and Measurement**, IEEE, v. 64, n. 3, p. 728–739, 2015. Citado na página 20.
- JOUAULT, F. **ATL/Tutorials - Create a simple ATL transformation**. 2013. Disponível em <https://wiki.eclipse.org/ATL/Tutorials_-_Create_a_simple_ATL_transformation> (accessado em 22/06/2020). Citado 2 vezes nas páginas 11 e 43.
- JÚNIOR, A. T. A. et al. Mobcons-at: uma ferramenta de autoria de restrições de mobilidade baseada na transformação de modelos. p. 122, jul 2018. Centro de Ciências Exatas e Tecnológicas. Disponível em: <<https://tedebc.ufma.br/jspui/handle/tede/tede/2446>>. Citado 4 vezes nas páginas 41, 82, 83 e 88.
- JáCOME, S.; JáCOME, M. S.; SÁNCHEZ-OCAÑA, W.; SALAZAR-PAREDES, R.; FERREIRA, J. Software development environments and tools in mde. **KnE Engineering**, v. 1, p. 1, 01 2018. Simposio Iberoamericano en Programación Informática. Citado na página 34.
- KARA, I. B. **Design and Implementation of the ModelicaML Code Generator Using Acceleo 3.X**. 55 p. Dissertação (Mestrado) — Linköping University Linköping University, Department of Computer and Information Science, The Institute of Technology, 2015. Linköping University, The Institute of Technology. Citado na página 48.
- KHALIFA, I. H.; KAMEL, A. E.; BARFETY, B. Real time indoor intelligent navigation system inside hypermarkets. **IFAC Proceedings Volumes**, v. 43, n. 8, p. 461 – 466, 2010. ISSN 1474-6670. 12th IFAC Symposium on Large Scale Systems: Theory and Applications. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1474667015334352>>. Citado na página 50.
- KHAN, A. M. **Model-based design for on-chip systems using and extending Marte and IP-XACT**. Tese (Theses) — Université Nice Sophia Antipolis, mar. 2010. Disponível em: <<https://tel.archives-ouvertes.fr/tel-00834283>>. Citado 2 vezes nas páginas 11 e 43.
- KIM, Y. J.; KANG, H. Y.; LEE, J. Development of indoor spatial data model using CityGML ADE. **International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives**, XL-2/W2, n. November, p. 41–45, 2013. ISSN 16821750. Citado na página 20.

- KOUHEN, A. E. **A metamodel specification for uml tools adaptation**. Tese (Theses) — Université des Sciences et Technologie de Lille - Lille I, dez. 2013. Disponível em: <<https://tel.archives-ouvertes.fr/tel-00997773>>. Citado 2 vezes nas páginas 12 e 68.
- KRUK, D. v. d. **A domain-specific language for specifying distributed real-time software system configurations**. Dissertação (Mestrado) — University of Twente, 2015. Faculty of Electrical Engineering, Mathematics and Computer Science. Citado 2 vezes nas páginas 14 e 38.
- KÜHNE, T. Matters of (meta-) modeling. **Software & Systems Modeling**, Springer Science and Business Media LLC, v. 5, n. 4, p. 369–385, jul. 2006. Disponível em: <<https://doi.org/10.1007/s10270-006-0017-9>>. Citado na página 33.
- Lan, G.; Junhui, H.; Jianqin, L.; Hui, W.; Jianzheng, D.; Jumou, Z. Research on substation gantry modeling based on ifc standard. In: **2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)**. [S.l.]: IEEE, 2019. p. 759–762. ISBN 978-1-7281-3584-7. Conference Location: Xiamen, China. Citado na página 29.
- LANO, K.; RAHIMI, S. K. High-integrity model-based development. p. 1–17, 01 2013. Citado na página 41.
- LEARNOSM. **Aprenda o OpenStreetMap Passo a Passo**. 2020. [Online; acessado em 16 de Julho de 2020]. Disponível em: <<https://learnosm.org/pt/josm/aerial-imagery/>>. Citado 2 vezes nas páginas 11 e 28.
- LEWIS, J. R. Measuring perceived usability: The CSUQ, SUS, and UMUX. **International Journal of Human-Computer Interaction**, Informa UK Limited, v. 34, n. 12, p. 1148–1156, jan. 2018. Disponível em: <<https://doi.org/10.1080/10447318.2017.1418805>>. Citado na página 88.
- Li, K.; Zlatanova, S.; Torres-Sospedra, J.; Perez-Navarro, A.; Laoudias, C.; Moreira, A. Survey on indoor map standards and formats. In: **2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)**. [S.l.: s.n.], 2019. p. 1–8. ISBN 978-1-7281-1788-1. Conference Location: Pisa, Italy. Citado na página 31.
- LI, K.-J.; LEE, J. Indoor spatial awareness initiative and standard for indoor spatial data. In: **Proceedings of IROS 2010 Workshop on Standardization for Service Robot**. [S.l.: s.n.], 2010. v. 18. Citado na página 19.
- Liao, M.; Wang, D.; Yang, H. Deploy indoor 2d laser slam on a raspberry pi-based mobile robot. In: **2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)**. [S.l.: s.n.], 2019. v. 2, p. 7–10. ISBN 978-1-7281-1859-8. Citado 4 vezes nas páginas 28, 57, 58 e 60.

- LIN, Z.; XIU, C.; YANG, W.; YANG, D. A graph-based topological maps generation method for indoor localization. In: **2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)**. [S.l.]: IEEE, 2018. p. 1–8. ISBN 978-1-5386-3755-5. Citado 2 vezes nas páginas 58 e 60.
- MARINHO, T. D. dos S. **Uma linguagem específica de domínio para geração de testes de performance**. 122 p. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, aug 2016. Mestrado Profissional em Engenharia de Software. Disponível em: <<https://repositorio.ufrn.br/jspui/handle/123456789/22312>>. Citado na página 82.
- MEDEIROS, A. L. **INavigS: uma infraestrutura de software ciente de contexto para navegação indoor**. 122 p. Dissertação (Mestrado) — Universidade Federal do Maranhão, aug 2018. Centro de Ciências Exatas e Tecnológicas. Disponível em: <<https://tedebc.ufma.br/jspui/handle/tede/tede/2408>>. Citado 7 vezes nas páginas 11, 12, 49, 50, 51, 54 e 62.
- MOLINA, J.; RUBIO, F.; PELECHANO, V.; VALLECILLO, A.; VARA, J.; VICENTE-CHICOTE, C. **Desarrollo de software dirigido por modelos: conceptos, métodos y herramientas**. [S.l.: s.n.], 2013. ISBN 978-84-9964-215-4. Citado na página 33.
- MUNIZ, L. C. **About the M-Hub/CDDL**. 2017. Available at <www.lsd.ufma.br/projetos/cddl/doku.php> (acessado em 13/04/2020). Citado na página 49.
- NEGASH, B.; WESTERLUND, T.; RAHMANI, A. M.; LILJEBERG, P.; TENHUNEN, H. Dos-ii: A domain specific internet of things language for resource constrained devices. **Procedia Computer Science**, Elsevier, v. 109, p. 416–423, 2017. Citado 2 vezes nas páginas 11 e 40.
- Ngui, M.; Lee, W. Low power wearable device with gps and indoor positioning system. In: **2019 International Conference on Green and Human Information Technology (ICGHIT)**. [S.l.: s.n.], 2019. p. 125–127. Citado na página 20.
- NOSSUM, A. S. Indoortubes a novel design for indoor maps. **Cartography and Geographic Information Science**, Taylor & Francis, v. 38, n. 2, p. 192–200, 2011. Citado 3 vezes nas páginas 11, 31 e 32.
- OCG. **Understanding IndoorGML**. 2019. [Online; acessado em 17 de Julho de 2020]. Disponível em: <<http://www.indoorgml.net/>>. Citado na página 30.
- OLIVEIRA, J. P. L. R. Thiago Araújo Silva de. **Geração de código estrutural implantável em nuvens a partir de modelos de componentes independentes de plataforma**. Dissertação (Mestrado) — Universidade Federal de Pernambuco, <https://repositorio.ufpe.br/handle/123456789/2811>, jan 2011. Citado na página 32.

- OSM. **Potlatch 2**. 2020. Disponível em <https://wiki.openstreetmap.org/wiki/Potlatch_2> (accessado em 22/05/2020). Citado na página 28.
- PADRINI-ANDRADE, L.; BALDA, R. d. C. X.; ARECO, K. C. N.; BANDIERA-PAIVA, P.; NUNES, M. d. V.; MARBA, S. T. M.; CARVALHO, W. B. d.; RUGOLO, L. M. S. d. S.; ALMEIDA, J. H. C. d.; PROCIANOY, R. S. et al. Avaliação da usabilidade de um sistema de informação em saúde neonatal segundo a percepção do usuário. **Revista Paulista de Pediatria**, SciELO Brasil, v. 37, n. 1, p. 90–96, 2019. Citado na página 96.
- PORTUGAL, I.; ALENCAR, P.; COWAN, D. A preliminary survey on domain-specific languages for machine learning in big data. In: IEEE. **Software Science, Technology and Engineering (SWSTE), 2016 IEEE International Conference on**. [S.l.], 2016. p. 108–110. Citado 2 vezes nas páginas 38 e 39.
- RIBEIRO, A.; SOUSA, L. D.; SILVA, A. Comparative analysis of workbenches to support dsmls: Discussion with non-trivial model-driven development needs. In: . [S.l.: s.n.], 2016. p. 323–330. Citado na página 67.
- ROCHA, J.; ALVES, N. Osm indoor: moving forward. In: **Open Source Geospatial Research & Education Symposium**. [S.l.: s.n.], 2012. p. 261–167. Citado na página 28.
- ROSA, A.; GONÇALVES, I.; PANTOJA, C. A mda approach for database modeling. **Lecture Notes on Software Engineering**, p. 26–30, jan 2013. Citado na página 44.
- ROSA, S. R. **A Rule-based Engine to support a Framework for the Experimental Validation of Domain Specific Languages**. Dissertação (Mestrado) — Universidade Nova de Lisboa, aug 2017. Centro de Ciências Exatas e Tecnológicas. Disponível em: <<http://hdl.handle.net/10362/30064>>. Citado 3 vezes nas páginas 72, 82 e 83.
- ROSER, H. R. M.; ORTIZ-OSPINA, E. World population growth. **Our World in Data**, 2013. <https://ourworldindata.org/world-population-growth>. Citado na página 19.
- RUBINO, I.; XHEMBULLA, J.; MARTINA, A.; BOTTINO, A.; MALNATI, G. Musa: Using indoor positioning and navigation to enhance cultural experiences in a museum. **Sensors (Basel, Switzerland)**, v. 13, p. 17445–71, 12 2013. Citado na página 50.
- SAKPERE, W.; OSHIN, M. A.; MLITWA, N. A state-of-the-art survey of indoor positioning and navigation systems and technologies. **South African Computer Journal**, v. 29, p. 145, 12 2017. Citado na página 19.

- SATAN, A. Bluetooth-based indoor navigation mobile system. In: IEEE. **2018 19th International Carpathian Control Conference (ICCC)**. [S.l.], 2018. p. 332–337. Citado na página 20.
- SCHÄFERMEIER, B.; HANIKA, T.; STUMME, G. Distances for wifi based topological indoor mapping. **arXiv preprint arXiv:1809.07405**, 2018. Citado na página 20.
- SNEPS-SNEPPE, M.; NAMIOT, D. On web-based domain-specific language for internet of things. In: IEEE. **Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2015 7th International Congress on**. [S.l.], 2015. p. 287–292. Citado na página 40.
- SOUSA, T. S. S. **Qualidade de Contexto em uma infraestrutura para Navegação Indoor**. 89 p. Dissertação (Mestrado) — Universidade Federal do Maranhão, apr 2019. Centro de Ciências Exatas e Tecnológicas. Disponível em: <<https://tedeuc.ufma.br/jspui/handle/tede/tede/2705>>. Citado 2 vezes nas páginas 49 e 52.
- SRIVASTAVA, S.; MAHESHWARI, N.; RAJAN, K. Towards generating semantically-rich indoorgml data from architectural plans. **ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**, XLII-4, p. 591–595, 09 2018. Citado 3 vezes nas páginas 29, 58 e 60.
- STAHL, T.; VÖLTER, M. **Model-Driven Software Development: Technology, Engineering, Management**. [S.l.: s.n.], 2006. 436 p. ISBN 9780470025703. Citado na página 42.
- TURNER, C.; LEWIS, J.; NIELSEN, J. Determining usability test sample size. **International Encyclopedia of Ergonomics and Human Factors**, Citeseer, v. 3, p. 3084–3088, 01 2006. Citado na página 101.
- VIANA, M. C. **Desenvolvimento e reuso de frameworks com base nas características do domínio**. 214 p. Tese (Doutorado) — Universidade Federal de Sao Carlos, may 2014. Centro de Ciências Exatas e Tecnológicas. Disponível em: <<https://repositorio.ufscar.br/handle/ufscar/291>>. Citado 2 vezes nas páginas 81 e 82.
- VIYOVIC, V.; MAKSIMOVIC, M.; PERISIC, B. Sirius: A rapid development of DSM graphical editor. In: **IEEE 18th International Conference on Intelligent Engineering Systems INES 2014**. IEEE, 2014. ISBN 978-1-4799-4615-0. Disponível em: <<https://doi.org/10.1109/ines.2014.6909375>>. Citado na página 46.
- WAGELAAR, D. **MMT/ATL Transformation Language (ATL)**. 2018. Disponível em <https://wiki.eclipse.org/ATL/Concepts> (accessado em 22/06/2020). Citado 2 vezes nas páginas 11 e 43.

- WANG, Z.; NIU, L. A data model for using openstreetmap to integrate indoor and outdoor route planning. **Sensors**, v. 18, p. 2100, 06 2018. Citado 3 vezes nas páginas 57, 58 e 60.
- WEN, C.; PAN, S.; WANG, C.; LI, J. An Indoor Backpack System for 2-D and 3-D Mapping of Building Interiors. **IEEE Geoscience and Remote Sensing Letters**, v. 13, n. 7, p. 992–996, 2016. ISSN 1545598X. Citado 3 vezes nas páginas 56, 58 e 60.
- YANG, L.; WORBOYS, M. Generation of navigation graphs for indoor space. **International Journal of Geographical Information Science**, Taylor & Francis, v. 29, n. 10, p. 1737–1756, 05 2015. Citado 2 vezes nas páginas 58 e 60.

A Questionários aplicados ao longo da pesquisa

A.1 Questionário Pré Experimento

Questionário Pré-Experimento

Gostaríamos de convidá-lo a participar de um estudo relacionado a pesquisa de desenvolvimento de uma Linguagem Específica de Domínio (DSL) para modelagem de modelos de representação do espaço indoor. Este estudo visa a avaliação de alguns aspectos da DSL proposta.

Antes de seguir para o preenchimento do formulário, leia com atenção e pergunte, caso tenha dúvida, antes de concordar na participação deste estudo.

O estudo está sendo conduzido por: Evaldo da Silva Andrade, aluno do curso de pós-graduação em ciência da computação da Universidade Federal do Maranhão (UFMA). Tendo como orientador o professor: Samyr Beliche Vale. Pesquisas apoiadas pelo Laboratório de Sistemas Distribuídos Inteligentes (LSDi).

Contexto: O objetivo deste survey é avaliar o perfil do entrevistado

Procedimentos: Se você aceitar participar neste estudo, será convidado a responder questões a respeito de seu conhecimento e habilidades em aspectos técnicos. As questões são de múltipla escolha. Deve levar entre 5-10 minutos para completar o survey. E enfatizando que sua participação é inteiramente voluntária.

Riscos: Ser um participante neste estudo não possui riscos previsíveis.

Benefícios: O pesquisador espera avaliar a DSL para modelagem de testes de desempenho comparado ao emprego das tradicionais abordagens para obtenção do modelo de representação do espaço indoor, por intermédio de ferramentas gráficas. Como participante, ao final do experimento, você poderá ter acesso aos resultados da pesquisa.

Confidencialidade: As informações e arquivos deste estudo serão classificadas como estritamente confidenciais, e por isso, deverão ser mantidas em privado. Qualquer tipo de publicação não deve conter qualquer informação que possibilite a identificação de nenhum dos participante. Apenas os pesquisadores primários deverão ter acesso aos arquivos.

Contato e perguntas: O pesquisador responsável pela condução do experimento é Evaldo da Silva Andrade, que pode ser contatado através do seguinte endereço de email: evaldo.andrade@ifto.edu.br. Ao redigir o email as perguntas devem conter a seguinte informação no campo assunto: "Questionamentos a respeito da DSL".

Caso esteja de acordo com termos, inicie respondendo às seguintes questões:

*Obrigatório

1. Endereço de email: *

1/4

Figura A.1 – Formulário pré experimento - página 1.

2. Nome: *

3. Nível Escolar *

Marcar apenas uma oval.

Graduação

Mestrado

Doutorado

4. Instituição onde realizou ou realiza a graduação. *

5. Atuação profissional

6. Conhecimento sobre programação de computadores. *

Marcar apenas uma oval.

Não tenho nenhum conhecimento

Conheço pouco

Já trabalhei com várias linguagens

7. Conhecimento sobre linguagens de programação. Se sim, Quais? *

8. Conhecimento sobre engenharia dirigida por modelos *

Marcar apenas uma oval.

Sim

Não

2/4

Figura A.2 – Formulário pré experimento - página 2.

9. Como você classifica seu conhecimento em Linguagens Específicas de Domínio (DSL)? *

Marcar apenas uma oval.

- Baixo, não possui qualquer conhecimento a respeito
- Regular, leitura de artigos, livros, cursos
- Médio, já trabalhei em pequenos projetos
- Alto, Já trabalhei em vários projetos em escala industrial.

10. Conhecimento sobre construção de modelos de representação do Espaço indoor. *

Marcar apenas uma oval.

- Baixo, não possuo qualquer conhecimento prévio
- Regular, tenho noção de como pode ser organizado os elementos em um modelo
- Médio, já realizei pequenos mapeamentos utilizando alguma ferramenta
- Alto, conheço as principais abordagens, modelos e ferramentas existentes

11. Conhecimento sobre engenharia dirigida por modelos *

Marcar apenas uma oval.

- Baixo, não possuo qualquer conhecimento prévio
- Regular, tenho conhecimento sobre o paradigma de desenvolvimento
- Médio, já trabalhei ou li a respeito e já realizei transformações em pequenos projetos
- Alto, conheço as principais abordagens e ferramentas existentes, e realizo trabalhos de forma frequente

3/4

Figura A.3 – Formulário pré experimento - página 3.

12. Conhecimento sobre ferramentas de modelagem (Ex. UML, EMF dentre outras)

*

Marcar apenas uma oval.

- Baixo, não possuo qualquer conhecimento prévio
- Regular, tenho conhecimento sobre o paradigma de desenvolvimento
- Médio, já trabalhei ou li a respeito e já realizei transformações em pequenos projetos
- Alto, conheço as principais abordagens e ferramentas existentes, e realizo trabalhos de forma frequente

Autorização

Concordo na participação do experimento respeitando as condições propostas. Garanto que irei realizar o experimento da melhor maneira possível e que todas as informações fornecidas são reais.

13. Estou de acordo com as informações acima preenchidas. *

Marcar apenas uma oval.

- Concordo
- Não concordo

4/4

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

Figura A.4 – Formulário pré experimento - página 4.

A.2 Questionário Pós Experimento

IndoorDSL Post-Experiment Evaluation

Post-Study System Usability Questionnaire.

*Obrigatório

1. Nome do participante *

2. No geral, estou satisfeito com o quão fácil é usar o sistema. *

Marcar apenas uma oval.

Concordo Fortemente Discordo Fortemente

3. Foi fácil utilizar este sistema. *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Concordo Fortemente Discordo Fortemente

4. Eu pude completar as tarefas e cenários de forma efetiva, usando este sistema. *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Concordo Fortemente Discordo Fortemente

5. Eu fui capaz de completar as tarefas e cenários de forma rápida, usando este sistema. *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Concordo Fortemente Discordo Fortemente

1/6

Figura A.5 – Formulário pré experimento - página 1.

6. Eu me senti confortável usando este sistema. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

7. *Marcar apenas uma oval.*

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

8. Eu acredito que eu poderia me tornar produtivo rapidamente usando este sistema. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

9. As mensagens de erros do sistema foram claras o suficiente para me ajudar na correção de erros. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

10. Sempre que eu cometi algum erro, eu pude recuperar de forma fácil e rápida. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

2/6

Figura A.6 – Formulário pré experimento - página 2.

11. As informações (apresentação do sistema, na tela de mensagens e outros documentos complementares) fornecidas com este sistema foram claras. *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Concordo Fortemente Discordo Fortemente

12. Foi fácil encontrar a informação que eu precisava. *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Concordo Fortemente Discordo Fortemente

13. A informação fornecida pelo sistema é fácil de entender. *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Concordo Fortemente Discordo Fortemente

14. A informação foi eficaz em me ajudar a completar as tarefas e cenários. *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Concordo Fortemente Discordo Fortemente

15. A organização das informações nas telas do sistema é clara. *

Marcar apenas uma oval.

1 2 3 4 5 6 7

Concordo Fortemente Discordo Fortemente

3/6

Figura A.7 – Formulário pré experimento - página 3.

16. A interface deste sistema é agradável. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

17. Eu gostei de usar a interface deste sistema. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

18. Este sistema tem todas as funções e capacidades que eu esperava que ele tivesse. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

19. Este sistema tem todas as funções e capacidades que eu esperava que ele tivesse. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

20. No geral, estou satisfeito com este sistema. *

Marcar apenas uma oval.

	1	2	3	4	5	6	7	
Concordo Fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Discordo Fortemente

4/6

Figura A.8 – Formulário pré experimento - página 4.

IndoorDSL System
Usability Scor

Nessa etapa, utilize os indicadores abaixo para mensurar sua resposta, são eles:
1 - Discordo fortemente, 2 - Discordo, 3 - Nem concordo ou discordo, 4 - Concordo, 5 - Concordo fortemente

21. Eu acho que gostaria de usar esse sistema com frequência *
- Marcar apenas uma oval.*
- 1 2 3 4 5
- Discordo fortemente Concordo fortemente
22. Eu acho o sistema desnecessariamente complexo. *
- Marcar apenas uma oval.*
- 1 2 3 4 5
- Discordo fortemente Concordo fortemente
23. Eu achei o sistema fácil de usar.
- Marcar apenas uma oval.*
- 1 2 3 4 5
- Discordo fortemente Concordo fortemente
24. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema. *
- Marcar apenas uma oval.*
- 1 2 3 4 5
- Discordo fortemente Concordo fortemente
25. Eu acho que as várias funções do sistema estão muito bem integradas. *
- Marcar apenas uma oval.*
- 1 2 3 4 5
- Discordo fortemente Concordo fortemente
26. Eu acho que o sistema apresenta muita inconsistência. *
- Marcar apenas uma oval.*
- 1 2 3 4 5
- Discordo fortemente Concordo fortemente

5/6

Figura A.9 – Formulário pré experimento - página 4.

27. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo fortemente Concordo fortemente

28. Eu achei o sistema atrapalhado (difícil) de usar. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo fortemente Concordo fortemente

29. Eu me senti confiante ao usar o sistema. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo fortemente Concordo fortemente

30. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema. *

Marcar apenas uma oval.

1 2 3 4 5

Discordo fortemente Concordo fortemente

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

6/6

Figura A.10 – Formulário pré experimento - página 4.

B Modelos desenvolvidos por meio da IndoorDSL referentes as tarefas propostas

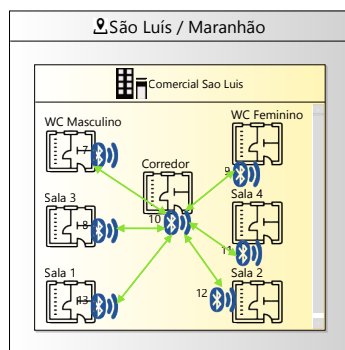


Figura B.1 – Modelo referente a tarefa fácil.

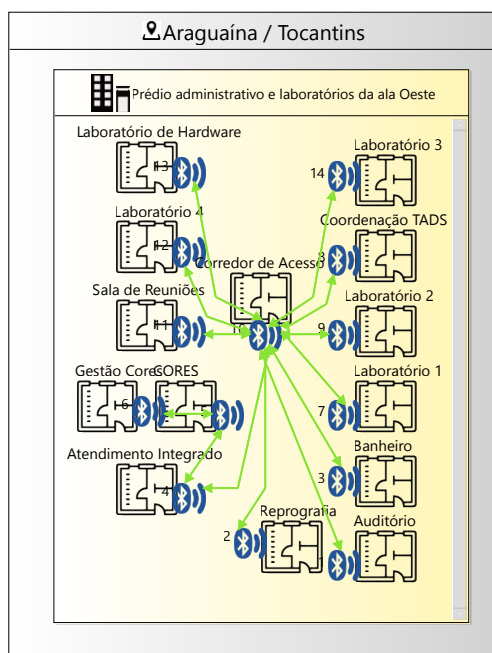


Figura B.2 – Modelo referente a tarefa fácil.

C Artefatos resultantes da transformação dos modelos elaborados nas tarefas.

```

1 /*
2 * import INavigS DAO (Data Access Object) classes
3 *
4 */
5 public class ObjectModel {
6     public static void (String[] args){
7         Address address1 = new Address(1, 'São Luís', 'Maranhão', '65045-
380', 'Avenida Edson Brandão', 156, 'lote especial', 'Anil');
8         Build build0 = new Build (1, 1, 'Comercial Sao Luis');
9
10        BuildElement buildElement1 = new BuildElement (1, 0, 'Banheiro',
'WC Masculino', RoomType.ENTRANCE, true, false);
11        BuildElement buildElement2 = new BuildElement (2, 0, 'Corredor de
acesso', 'Corredor', RoomType.ROOM, true, false);
12        BuildElement buildElement3 = new BuildElement (3, 0, 'Sala de
aula', 'Sala 3', RoomType.PATH, true, false);
13        BuildElement buildElement4 = new BuildElement (4, 0, 'Banheiro',
'WC Feminino', RoomType.ROOM, true, false);
14        BuildElement buildElement12 = new BuildElement (12, 0, 'Sala de
aula', 'Sala 1', RoomType.ROOM, true, false);
15        BuildElement buildElement13 = new BuildElement (13, 0, 'Sala de
aula', 'Sala 4', RoomType.ROOM, true, false);
16        BuildElement buildElement14 = new BuildElement (14, 0, 'Sala de
aula', 'Sala 2', RoomType.ROOM, true, false);
17        Beacon beacon7 = new Beacon (7, 1, 'WCM1', '6f14bef9-
406b-476e-8df5-b0f74bd319f3', 'Banheiro', -2.556346, -44.242318);
18        Beacon beacon10 = new Beacon (10, 2, 'C01', '67542ae5-
0baf-44cf-872d-e3fbf78acld7', 'Corredor', -2.556352, -44.242252);
19        Beacon beacon8 = new Beacon (8, 3, 'S03', '5b1b6b6d-0b79-
4c4b-bef2-ef171987ccaf', 'Sala', -2.556328, -44.242289);
20        Beacon beacon9 = new Beacon (9, 4, 'WCF1', '786dba59-1b51-
42bb-b7f7-80d69b5a8d54', 'Banheiro', -2.556412, -44.242262);
21        Beacon beacon13 = new Beacon (13, 12, 'S01', '89b3a897-fd3f-4c27-
820b-5e5a58672e58', 'Sala', -2.556292, -44.242258);
22        Beacon beacon11 = new Beacon (11, 13, 'S04', '17de67b9-f5c0-
4ba0-b3f0-320dc26837fb', 'Sala', -2.55639, -44.242235);
23        Beacon beacon12 = new Beacon (12, 14, 'S02', 'b4ce7036-7ab8-
4000-b75e-8c2574b07892', 'Sala', -2.556356, -44.242199);
24        Link link1 = new Link (12, 10, 1, 274.29155304925246, 4.0);
25        Link link2 = new Link (11, 10, 2, 335.77562516463615, 4.0);
26        Link link3 = new Link (9, 10, 3, 9.515499005793869, 4.0);
27        Link link4 = new Link (13, 10, 4, 174.25693599058704, 4.0);
28        Link link5 = new Link (8, 10, 5, 122.82026724772959, 4.0);
29        Link link6 = new Link (7, 10, 6, 95.16502584426, 4.0);
30    }
31 }

```

Figura C.1 – Código gerado para o modelo referente a tarefa fácil.

```

1 /*
2 * import INavigS DAO (Data Access Object) classes
3 *
4 */
5 public class ObjectModel {
6     public static void (String[] args){
7         Address address1 = new Address(1, 'Araguaína', 'Tocantins', '77824
-838', 'Avenida Paraguai, esquina com Avenida Amazonas, Quadra 56', 1,
'Prédio anexo I', 'Cimba');
8         Build build1 = new Build (1, 1, 'Prédio administrativo e
laboratórios da ala Oeste');
9
10        BuildElement buildElement1 = new BuildElement (1, 1, 'Laboratório
de Hardware', 'Laboratório de Hardware', RoomType.ROOM, true, false);
11        BuildElement buildElement2 = new BuildElement (2, 1, 'Laboratório
3', 'Laboratório 3', RoomType.ROOM, true, false);
12        BuildElement buildElement3 = new BuildElement (3, 1, 'Coordenação
TADS', 'Coordenação TADS', RoomType.ROOM, true, false);
13        BuildElement buildElement4 = new BuildElement (4, 1, 'Laboratório
2', 'Laboratório 2', RoomType.ROOM, true, false);
14        BuildElement buildElement5 = new BuildElement (5, 1, 'Laboratório
1', 'Laboratório 1', RoomType.ROOM, true, false);
15        BuildElement buildElement6 = new BuildElement (6, 1, 'Banheiro',
'Banheiro', RoomType.ROOM, true, false);
16        BuildElement buildElement7 = new BuildElement (7, 1, 'Auditório',
'Auditório', RoomType.ROOM, true, false);
17        BuildElement buildElement8 = new BuildElement (8, 1, 'Atendimento
Integrado', 'Atendimento Integrado', RoomType.ROOM, true, false);
18        BuildElement buildElement9 = new BuildElement (9, 1, 'CORES',
'CORES', RoomType.ROOM, true, false);
19        BuildElement buildElement10 = new BuildElement (10, 1, 'Gestão
Cores', 'Gestão Cores', RoomType.ROOM, true, false);
20        BuildElement buildElement11 = new BuildElement (11, 1,
'Laboratório 4', 'Laboratório 4', RoomType.ROOM, true, false);
21        BuildElement buildElement12 = new BuildElement (12, 1, 'Sala de
Reuniões', 'Sala de Reuniões', RoomType.ROOM, true, false);
22        BuildElement buildElement13 = new BuildElement (13, 1,
'Reprografia', 'Reprografia', RoomType.ROOM, true, false);
23        BuildElement buildElement14 = new BuildElement (14, 1, 'Corredor
de Acesso', 'Corredor de Acesso', RoomType.ROOM, true, false);
24        Beacon beacon13 = new Beacon (13, 1, 'HS005', '92962f56-c5a6-
4e9d-9b05-0d63b9be764a', 'Laboratório de Hardware', -7.181305,
-48.193759);
25        Beacon beacon14 = new Beacon (14, 2, 'HS003', '7cd5ce8b-03a4-
43d5-a23e-3eb1848073c9', 'Laboratório 3', -7.181304, -48.193678);
26        Beacon beacon8 = new Beacon (8, 3, 'HS004', 'c589dca5-d154-
451f-929f-85c275a5aab5', 'Coordenação TADS', -7.181356, -48.193681);
27        Beacon beacon9 = new Beacon (9, 4, 'HN002', '7c630dfc-7ba4-
4cdb-8ecc-771d7a3ff703', 'Laboratório 2', -7.181395, -48.193685);
28        Beacon beacon7 = new Beacon (7, 5, 'SS004', '1f34b639-7007-4672-
8a64-b85b1655aaf1', 'Laboratório 1', -7.181469, -48.193695);
29        Beacon beacon3 = new Beacon (3, 6, 'SN004',
'6b7ce332-b67a-4aa4-bcd6-a9dfd2ff6e77', 'Banheiro', -7.181569,
-48.193708);
30        Beacon beacon1 = new Beacon (1, 7, 'BN001', '7f7f823d-82f7-
4485-a8e3-81f6fc74ble1', 'Auditório', -7.181542, -48.193646);
31        Beacon beacon4 = new Beacon (4, 8, 'CN003', 'b2e8f5d1-8176-
4fc6-b8d8-12647df94363', 'Atendimento integrado', -7.181484,
-48.193772);
32        Beacon beacon5 = new Beacon (5, 9, 'C1S001', 'aefec15b-e7c2-4ff0-
8a0e-2537780213f0', 'CORES', -7.181463, -48.193746);
33        Beacon beacon6 = new Beacon (6, 10, 'RS001', '77872871-f8b3-
4007-a6d3-efc826c6f3f3', 'Gestão Cores', -7.181462, -48.193781);
34        Beacon beacon12 = new Beacon (12, 11, 'HN004', 'ac2ba531-
821e-42f9-a269-e2f0f28d0b91', 'Laboratório 4', -7.181359, -48.193765);
35        Beacon beacon11 = new Beacon (11, 12, 'HS006', 'a2527e47-8b63-
444b-a774-0df1be9efa98', 'Sala de Reuniões', -7.181405, -48.193764);
36        Beacon beacon2 = new Beacon (2, 13, 'SN001',
'df31048e-74fa-4006-b1f5-ac5a0e0d1924', 'Reprografia', -7.181569,
-48.19372);
37        Beacon beacon10 = new Beacon (10, 14, 'HS007', '2b240c77-157b-4b56
-8b94-81b59a714869', 'Corredor de Acesso', -7.181429, -48.193724);
38        Link link1 = new Link (14, 10, 1, 83.67753478419513, 4.0);
39        Link link2 = new Link (13, 10, 2, 83.67755182884012, 4.0);
40        Link link3 = new Link (8, 10, 3, 83.67749026759792, 4.0);
41        Link link4 = new Link (12, 10, 4, 83.67750623814463, 4.0);
42        Link link5 = new Link (9, 10, 5, 83.67745726718957, 4.0);
43        Link link6 = new Link (11, 10, 6, 83.67746605013797, 4.0);
44        Link link7 = new Link (7, 10, 7, 83.67739518412601, 4.0);
45        Link link8 = new Link (5, 4, 8, 231.03706375865866, 4.0);
46        Link link9 = new Link (6, 5, 9, 91.63864717395136, 4.0);
47        Link link10 = new Link (4, 10, 10, 83.67739918071288, 4.0);
48        Link link11 = new Link (3, 10, 11, 83.67731117463514, 4.0);
49        Link link12 = new Link (2, 10, 12, 83.67731382858094, 4.0);
50        Link link13 = new Link (1, 10, 13, 83.67732092144324, 0.0);
51    }
52 }

```

Figura C.2 – Código gerado para o modelo referênte a tarefa difícil.