
Estudo para a Modelagem de um Sistema Moderno por meio da UML e extensões

Igor Henrique Correia Resende



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Monte Carmelo - MG
2019

Igor Henrique Correia Resende

**Estudo para a Modelagem de um Sistema
Moderno por meio da UML e extensões**

Trabalho de Conclusão de Curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau de
Bacharel em Sistemas de Informação.

Área de concentração: Ciência da Computação

Orientador: Thiago Pirola Ribeiro

Monte Carmelo - MG
2019

Gostaria de dedicar esse trabalho principalmente à Deus por ser tão presente e essencial em minha vida, o autor do meu destino, meu guia que nunca me abandonou, aos meus Professores, Familiares, minha namorada e Amigos, por ter me dado apoio e companheirismo ao longo do tempo.

Agradecimentos

Agradeço primeiramente a Deus por não me deixar desistir desta jornada, e por estar presente em todo momento da minha vida.

Agradeço a minha família, pela capacidade de acreditar e investir em mim. A minha mãe, Márcia, por seu cuidado e dedicação que foi me dado, a sua palavra amiga nas horas que pensei em desistir. Ao meu pai, Paulo, sua presença significou segurança e certeza de que não estou sozinho nessa batalha, aos meus irmãos Diego e Thyago que foram essenciais com o apoio e incentivo de cada um, e a minha namorada Andressa que me deu força e coragem, apoiando nos momentos de dificuldades, sem ela eu não teria conseguido.

Dedico todo o esforço que depositei neste trabalho as minhas avós, especialmente ao meu avô Benedito José (in memorian), que sempre me apoiou e se preocupou comigo.

A todos os professores do Curso de Sistemas de Informação, principalmente ao meu professor orientador Thiago Pirola, pois sem seus ensinamentos e conselhos não seria capaz de concluir mais essa jornada na minha vida.

“Só se pode alcançar um grande êxito quando nos mantemos fiéis a nós mesmos.”
(Friedrich Nietzsche)

Resumo

Atualmente inúmeros sistemas são projetados diariamente, dentre eles, aplicações WEB, aplicações Mobile, aplicações para Internet das Coisas, aplicações para Armaazenamento em Nuvem, dentre outras, evidenciando a carência que se tem na integridade e exibições entre as várias modelagens publicadas. Com isso, a realização de uma análise se fez necessária para o indicativo das melhores práticas de modelagens e utilização dos seus componentes, tal qual, ligação entre estereótipos, extensões, elementos que possam ser definidos e integrados para serem utilizados em aplicações específicas ou em conjunto. Após essa análise, gerou-se um conjunto de indicativos exemplificados em diagramas que englobam as diversas aplicações atuais.

Palavras-chave: UML, Modelagem, Estereótipos, Sistemas Modernos.

Lista de ilustrações

Figura 1 – Tendências globais no desenvolvimento de aplicações em IoT. Modificado de Scully (2018).	17
Figura 2 – Exemplo de um modelo simples de <i>EntityType</i> . Fonte: Pires et al. (2015).	18
Figura 3 – Exemplo de um modelo hierárquico de <i>EntityType</i> . Fonte: Pires et al. (2015).	19
Figura 4 – Parte dos Diagramas da UML 2.0. Fonte: Martinez (2007)	23
Figura 5 – Ícones dos estereótipos: (a) «server page», (b) «client page», (c) «HTML Form», (d) «static page», (e) «dynamic page», (f) «script library» de classe e (g) «script library» de componente. Fonte: Oliveira e Vahldick (2010).	25
Figura 6 – Diagrama de Classe	27
Figura 7 – Diagrama Caso de Uso	27
Figura 8 – Diagrama de Objetos	28
Figura 9 – Diagrama de Componentes	29
Figura 10 – Diagrama de Pacotes	29
Figura 11 – Diagrama de Atividades	30
Figura 12 – Diagrama de Sequências	30
Figura 13 – Diagrama de Implantação	31
Figura 14 – Diagrama de Estado	32
Figura 15 – Diagrama de Comunicação	32
Figura 16 – Diagrama de Estrutura Composta	33
Figura 17 – Diagrama de Interatividade	33
Figura 18 – Diagrama Web - WAE	34
Figura 19 – Diagrama IoT	34
Figura 20 – Diagrama Cloud	34
Figura 21 – Ideia geral do projeto a ser modelado. Fonte: (SILVA, 2019)	35
Figura 22 – Diagrama Sequência - Login Aluno	36
Figura 23 – Diagrama Sequência (Moodle) - Login Professor	37

Figura 24 – Diagrama Sequência (USB) - Login Professor	38
Figura 25 – Diagrama Sequência (CLOUD) - Login Professor	39
Figura 26 – Diagrama Caso de Uso - Aluno	39
Figura 27 – Diagrama Caso de Uso - IoT	40
Figura 28 – Diagrama Caso de Uso - Professor	41
Figura 29 – Diagrama Caso de Uso - SmartPhone	42
Figura 30 – Diagrama Caso de Uso - Cloud	42
Figura 31 – Diagrama Caso de Uso - Servidores	43
Figura 32 – Estereótipos na notação WAE	43
Figura 33 – Representação Login no Sistema - WAE.	44
Figura 34 – Sugestões de novos ícones dos estereótipos WAE: (a) «device IoT», (b) «Cloud», (c) «Servers», (d) «device mobile» (e) «Sockets Comunica- ção», (f) «Projetor Multimídia».	45
Figura 35 – Modelagem WAE Proposta - Sala de Aula Inteligente com IoT.	45

Lista de tabelas

Tabela 1 – Diagramas Estruturais da UML. Modificado de (GROFFE, 2013) . . .	22
Tabela 2 – Diagramas Comportamentais da UML. Modificado de (GROFFE, 2013). .	22
Tabela 3 – Diagramas de Interação da UML. Modificado de (GROFFE, 2013) . . .	23
Tabela 4 – Combinação de Estereótipos com relacionamentos válidos.	44

Lista de siglas

EC2 Nuvem de Computação Elástica - *Elastic Compute Cloud*

EV Entidades Virtuais - *Virtual Entity*

HDMI Interface Multimídia de Alta Definição - *High-Definition Multimedia Interface*

IEEE Instituto de Engenheiros Eletricistas e Eletrônicos - *Institute of Electrical and Electronics Engineers*

IETF Força Tarefa de Engenharia da Internet - *Internet Engineering Task Force*

IoT Internet das Coisas - *Internet Of Things*

ITU União Internacional de Telecomunicações - *International Telecommunication Union*

LAN Rede de Área Local - *Local Area Network*

OWL Linguagem de Ontologia para Web - *Web Ontology Language*

S3 Solução de Armazenamento Simples - *Simple Storage Solution*

SGBD Sistema de Gerenciamento de Banco de Dados - *Database Management System*

UML Linguagem de Modelagem Unificada - *Unified Modeling Language*

WAE Extensão de Aplicativo para Web - *Web Application Extension*

WEB Rede Mundial de Computadores - *World Wide Web*

Sumário

1	INTRODUÇÃO	11
1.1	Motivação	13
1.2	Objetivos	13
1.3	Contribuições	14
1.4	Organização da Monografia	14
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	<i>Mobile</i>	15
2.2	Internet das Coisas	16
2.3	Entidade Virtuais	18
2.4	Computação em Nuvem	18
2.5	Sistemas Distribuídos	20
2.6	UML	21
3	EXPERIMENTOS E ANÁLISE DOS RESULTADOS	26
3.1	Experimentos	26
3.1.1	Modelagem de um Sistema Moderno	35
3.1.2	Modelando com Casos de Uso	37
3.1.3	Extensões UML para Aplicações Web	38
3.2	Avaliação dos Resultados	45
4	CONCLUSÃO	47
4.1	Principais Contribuições	48
4.2	Trabalhos Futuros	48
4.3	Contribuições em Produção Bibliográfica	48
	REFERÊNCIAS	49

CAPÍTULO 1

Introdução

Na atualidade, com a agitada vida moderna, é quase impossível a sobrevivência sem a utilização de dispositivos tecnológicos, pois a exigência para ter desempenho mais rápidos das funções, tanto na vida doméstica quanto na profissional. É notório que encontram-se máquinas capazes de aceitar dados de entradas, processá-los e gerar um resultado em todo o nosso redor, sendo grandes exemplos: calculadora, televisão, blu-ray, aparelhos reprodutores de sons, micro-ondas, aparelhos para auxiliar na área medicinal, dentre outros.

O desenvolvimento dessas máquinas surgiu para auxiliar nos processos de cálculos aritméticos, como nas tarefas repetitivas devido a necessidade de resultados rápidos e exatos nas mais diversas áreas da humanidade, na medida em que os cálculos foram ficando mais complexos e aumentando de tamanho.

O ser humano tomou o rumo das máquinas de processar dados somente depois que dominou a escrita e inventou o papel e nunca mais parou com suas fantásticas criações. E atualmente a sociedade sofre um grande impacto dessa tecnologia, provocando uma dependência cada vez maior.

Hoje em dia, os computadores e celulares são usados tanto para lazer, trabalho ou qualquer outra atividade e, tudo isso, graças à evolução da informática moderna.

Uma vez que a evolução das máquinas não param, foi possível notar a grande importância de se ter bons softwares modelados pois, a modelagem é uma das principais atividades que levam à implementação de um bom software. Construir modelos para comunicar a estrutura e o comportamento desejado do sistema, visualizar e controlar a arquitetura do mesmo e compreender melhor o sistema que se está elaborando é muito desejável.

A modelagem de software utiliza diversos modelos para projetar um determinado sistema. Um modelo é uma simplificação da realidade, criado para facilitar o entendimento de sistemas complexos. Estes modelos podem abranger planos detalhados, assim como planos mais gerais com uma visão panorâmica do sistema.

Todos os sistemas podem ser descritos sob diferentes aspectos, com a utilização de

modelos distintos, onde cada modelo será, portanto, uma abstração específica do sistema. Os modelos podem ser estruturais, dando ênfase à organização do sistema, ou podem ser comportamentais, dando ênfase à dinâmica do sistema.

De acordo com Booch, Rumbaugh e Jacobson (2006), há quatro objetivos principais para se criar modelos:

1. Ajudam a visualizar o sistema como ele é ou como se deseja que ele seja;
2. Permitem especificar a estrutura ou o comportamento de um sistema;
3. Proporcionam um guia para a construção do sistema;
4. Documentam as decisões tomadas no projeto.

Através dos modelos, consegue-se obter múltiplas visões do sistema, particionando a complexidade do sistema para facilitar sua compreensão, e atuando como meio de comunicação entre os participantes do projeto. Portanto, uma linguagem de modelagem padronizada, tal como a UML, é fundamental para a construção e o entendimento de bons modelos.

A grande quantidade de aplicações que se tem na área de tecnologia, como aplicações: WEB, Móveis, Internet das Coisas (IoT), Computação em Nuvem (*Cloud*), Sistemas Distribuídos, dentre outras, a modelagem desses sistemas antes da implementação vem sendo um grande problema para os profissionais da área.

O aumento da mobilidade de dispositivos tem gerado a necessidade de novas aplicações que permitam o acesso a informações de forma eficiente no ambiente de comunicação sem fio, então surgindo a necessidade de modelar serviços para esse tipo de ambiente.

Grande parte das aplicações citadas anteriormente, são modeladas com um único recurso que é a Linguagem de Modelagem Unificada (UML), uma linguagem de modelagem visual orientada a objetos. As linguagens de modelagem orientada a objetos surgiram entre a metade da década de 1970 e o final da década de 1980. Com isso os desenvolvedores começaram a experimentar outros métodos de análise e projetos a partir das experiências adquiridas com a programação orientada a objetos.

Visto a dificuldade que se encontrava para modelar com UML foi necessário a criação de algumas extensões para UML onde a mesma poderia ser adaptada para desenvolvimento de algumas aplicações. Diante disso, acrescentou-se a elas algumas novas extensões a fim de modelar as limitações relativas à interface gráfica e interação com o usuário na mobilidade, melhorando dessa forma, a usabilidade desses sistemas.

A UML possui elementos simples para modelagem orientada a objetos, por exemplo: casos de uso, classes, dentre outros. A UML fornece os principais elementos para modelagem de um sistema, porém para a modelagem de um sistema que envolve restrições e limitações, alguns elementos extras são necessários.

Implementar softwares com grande aprovação vindo dos usuários é o principal motivo dos estudos em usabilidade de software. Uma possível maneira de melhorar a usabilidade em sistemas móveis é a criação de extensões UML com o propósito de antecipar e evitar problemas da interação com o usuário já na fase de modelagem do software, permitindo, assim, um projeto mais cuidadoso e o consequente aumento de sua qualidade na implementação final pelo desenvolvedor.

A qualidade em desenvolvimento de software é um dos assuntos mais discutidos na área acadêmica, gerando várias pesquisas e discussões sobre esse tópico (KOSCIANSKI; SOARES, 2007). Tudo isso devido a maior demanda por softwares que fazem exatamente as tarefas que se propõem a fazer, porém de maneira eficaz.

Nesse sentido, este trabalho tem como objetivo ajudar auxiliar por meio de indicativos, sugerir uma modelagem que se adapta a determinada aplicação, e demonstrar a dificuldade que se tem quando se une diversas aplicações em um único sistema.

1.1 Motivação

Com a grande quantidade de sistemas computacionais que se tem atualmente, bem como sistemas em plataformas diversificadas (IoT, *Cloud*, *Mobile*, etc) com isto se tem grande dificuldade para realizar a modelagem antes da implementação dessas plataformas, demandando o uso de pesquisas em materiais para estar auxiliando a melhor modelagem, dessa forma para facilitar desde o levantamento de requisito até o processo de todo implementação do sistema.

No trabalho desenvolvido e implementado em (SILVA, 2019), envolveu diversas plataformas e, percebeu-se a dificuldade na modelagem das aplicações implementadas, visto que não foram identificados modelos apropriados para as implementações realizadas.

1.2 Objetivos

O objetivo deste trabalho é gerar indicativos da utilização dos diferentes tipos de modelagens e extensões para o desenvolvimento de aplicações, apresentando as diferenças de cada modelagem, auxiliando na modelagem desde o levantamento de requisitos até o projeto final de sua implementação. Os objetivos específicos do trabalho são:

- ❑ Analisar a utilização da modelagem em determinados “tipos de aplicações”.
- ❑ Identificar as extensões desenvolvidas para cada “tipo de aplicações”.
- ❑ Avaliar individualmente as modelagens e apresentar um comparativo entre essas.
- ❑ Modelar um sistema composto por diversos “tipos de aplicações”.

1.3 Contribuições

Após gerar os diagramas e explicar cada componente dos mesmos, espera-se que seja possível dar uma visão clara e ampla da utilização dos diagramas, podendo assim, auxiliar os desenvolvedores sobre qual modelagem utilizar na implementação de um sistema específico.

1.4 Organização da Monografia

O Capítulo 2 abordará a fundamentação teórica com o objetivo de auxiliar na compreensão do trabalho. No Capítulo 3 serão apresentados os experimentos e análise dos resultados, que abrangerão o desenvolvimento durante o trabalho. Finalizando com um Capítulo englobando as conclusões juntamente com as principais contribuições obtidas no trabalho e indicativos de trabalhos futuros.

CAPÍTULO 2

Fundamentação Teórica

Este capítulo tratará de algumas aplicações disponíveis no mercado de tecnologia, e mostrará os obstáculos das mesmas para aplicar a modelagem e, também, demonstrará a que se refere cada aplicação e seu uso.

2.1 *Mobile*

Segundo o dicionário Michaelis (2006) *mobile* refere-se ao termo “móvel”, porém levando em consideração a área de tecnologia, *mobile* pode ser definido como um “computador” portátil, com tela de tamanho reduzido e de espessura fina.

Computação móvel é uma tecnologia que permite que as pessoas obtenham serviços computacionais independente de sua localização física. A fim de facilitar a mobilidade do usuário, os dispositivos utilizados são pequenos, com o intuito de facilitar sua locomoção. Além de ser um paradigma computacional que tem como objetivo prover ao usuário acesso permanente a uma rede independente de sua posição física (FIGUEIREDO; NAKAMURA; FIGUEIREDO, 2015).

O uso de equipamentos portáteis que utilizam tecnologia de comunicação sem fio vem alterando a maneira como as pessoas interagem com informações e serviços que antes só eram acessados por meio de computadores de mesa (*Desktops*). Com o surgimento desses novos equipamentos, aplicações e serviços estão surgindo para atender às necessidades do usuário móvel (CYBIS; BETIOL; FAUST, 2007).

Utiliza-se tecnologia móvel para acessar a Internet e outros recursos computacionais por meio de dispositivos móveis: celulares, iPhone, iPod, iPad, tablets, notebooks, laptops, smartpads, entre outros. Cada vez mais os dispositivos móveis se tornam integrados e utilizados por um número maior de pessoas devido fácil acesso as informações em qualquer lugar.

Com o aumento dos aparelhos móveis, é notório que existe um grande interesse em se desenvolver serviços e aplicações cada vez mais elaboradas e com alto grau de qualidade. Dessa maneira, um grande número de pesquisadores tem trabalhado no sentido de

procurar soluções que atendam e melhorem os serviços móveis, e muitas companhias têm investido no desenvolvimento de dispositivos que atendam às necessidades dos usuários.

Muitos trabalhos relacionados à usabilidade em sistemas móveis têm sido desenvolvidos com o intuito de facilitar a navegação dos usuários através dos diversos dispositivos (JONES et al., 1999). Estudos de usabilidade em aplicações em dispositivos móveis devem ser centrados nas restrições de *design* impostas por uma limitação de banda e pelos pequenos *displays* dos dispositivos (CHAN et al., 2002).

2.2 Internet das Coisas

A Internet das Coisas (IoT) é uma coleção de padrões e tecnologias que, apesar de muito estudado, ainda não possui uma definição exata. Órgãos como o Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE), Força Tarefa de Engenharia da Internet (IETF) e União Internacional de Telecomunicações (ITU) declaram as seguintes definições para o termo IoT:

- ❑ IEEE: “Uma rede de itens - cada um incorporado com sensores - que estão conectados à Internet” (IEEE, 2015);
- ❑ IETF: “A ideia básica da IoT é conectar objetos à nossa volta (eletrônicos, elétricos, não elétricos) para prover comunicação transparente e serviços” (IEEE, 2014);
- ❑ ITU: “Uma rede dinâmica que conecta redes a qualquer hora, qualquer lugar, com conexão para qualquer pessoa. Têm-se agora conectividade para todas as coisas” (IEEE, 2014).

De acordo com Oliveira (2017), o termo IoT está muito além do que apenas ligar lâmpadas pelo celular. Não é apenas isso, é tornar inteligentes, capazes de coletar e processar informações do ambiente ou das redes às quais estão conectadas. O conceito não é atual, pois antigamente já se imaginava maneiras de interligar os equipamentos utilizados no dia-a-dia com a Internet. Algumas tecnologias possibilitaram essa comunicação através de sensores e softwares.

Internet das Coisas tem como promessa revolucionar a maneira como as pessoas vivem, trabalham e interagem, oferecendo novas e inovadoras maneiras para as organizações gerenciarem e monitorarem operações remotas. São inúmeras possibilidades de uso, desde algumas soluções que costumeiras até outras inimagináveis. Pode-se usufruir da Internet das Coisas até para utilidade pública: inúmeros serviços públicos como polícia, bombeiros, ambulância e outros, podem ser aprimorados em termos de agilidade com a ajuda de dispositivos conectados às suas redes.

A Internet das Coisas tem capacidade para englobar aplicações nas mais diversas áreas, desde casas conectadas até cidades inteligentes. De acordo com a *IoT Analytics*, as

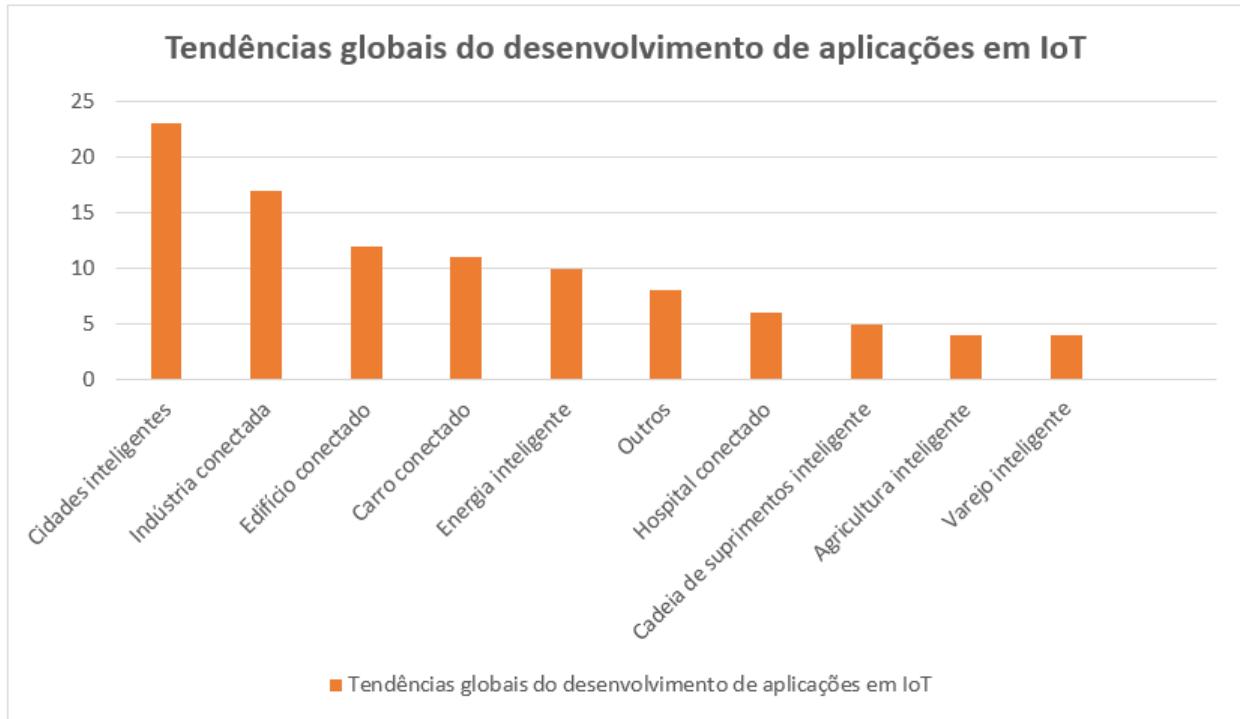


Figura 1 – Tendências globais no desenvolvimento de aplicações em IoT. Modificado de Scully (2018).

quatro principais tendências globais em aplicações da Internet das Coisas no ano de 2016 foram: indústrias conectadas, *smart city* - (cidade inteligente), *smart energy* - (energia inteligente) e carros conectados. Porém, conforme ilustra a Figura 1, nas tendências em 2018 percebe-se que algumas se mantiveram e outras se modificaram, que é o caso de edifícios conectados.

Pode-se, também, utilizar IoT para organização pessoal, automação de tarefas diárias, controle financeiro, contratação de serviços, entre outras tantas atividades focadas no dia-a-dia. Alguns exemplos do que se pode fazer com Internet das Coisas:

- ❑ Casas Inteligentes - Amazon Echo (SILVA, 2014);
- ❑ Dispositivos vestíveis - Fitbit (ALHARBI et al., 2016);
- ❑ Cidades Inteligentes - Barcelona (URBANHUB, 2018);
- ❑ Veículos conectados - Apple CarPlay (CARPLAY, 2018);

Objetos inteligentes conectados à IoT têm como objetivo enfatizar a troca de informações entre eles e também com sistemas externos. Porém, a forma como definir, estruturar, armazenar, processar, gerenciar e trocar as informações é de grande valia. Dessa maneira, a visão da informação permite visualizar estruturas de informações estáticas e fluxo de informações dinâmicas. Essa visão possui enfoque na descrição, tratamento e ciclo de vida da informação, bem como no fluxo de informações através do sistema e os componentes

envolvidos. Com isso, a próxima seção ilustrará um ponto de vista somente da modelagem de Entidades Virtuais (EV).

2.3 Entidade Virtuais

Entidades Virtuais (EV) são as definições chaves para qualquer sistema de IoT, no qual modelar a entidade física (objeto) é o ponto de interesse real. Uma EV tem um identificador, um tipo (*EntityType*) e atributos que fornecem informação sobre a EV ou podem ser utilizados para trocar o seu estado, provocando um estímulo sobre a entidade física modelada. A modelagem do *EntityType* é muito importante uma vez que este pode ser utilizado para determinar quais atributos uma instância de EV pode ter, definindo assim sua semântica. Um *EntityType* pode ser modelado como um elemento simples (Figura 2) e de forma hierárquica (Figura 3).



Figura 2 – Exemplo de um modelo simples de *EntityType*. Fonte: Pires et al. (2015).

EntityTypes são similares as classes na programação orientada a objetos. Sendo assim, diagramas de classes UML podem ser utilizados para modelar *EntityTypes*. Em particular, a relação de generalização pode ser usada para modelar subclasses de *EntityTypes*, conforme mostrado na Figura 3. Além da UML, linguagens de ontologias como a Linguagem de Ontologia para Web (OWL) podem ser usadas de maneira alternativa, uma vez que fornecem meios para modelar classes e subclasses.

2.4 Computação em Nuvem

Segundo Taurion (2009), Computação em Nuvem (*Cloud Computing*) é a ideia de utilizar, seja em qual for o lugar e independente de plataforma, os mais variados tipos de aplicações através da Internet com a mesma facilidade de tê-las instaladas localmente, no próprio computadores.

Armazenar arquivos em servidores que podem ser acessados a qualquer hora, ficando o processamento na "nuvem", necessitando o usuário apenas de um navegador ou aplicativo e uma conexão com Internet para utilizar o serviço. Exemplos de utilização deste tipo de armazenamento em nuvem são os serviços da Amazon (AMAZON, 2018b), Dropbox

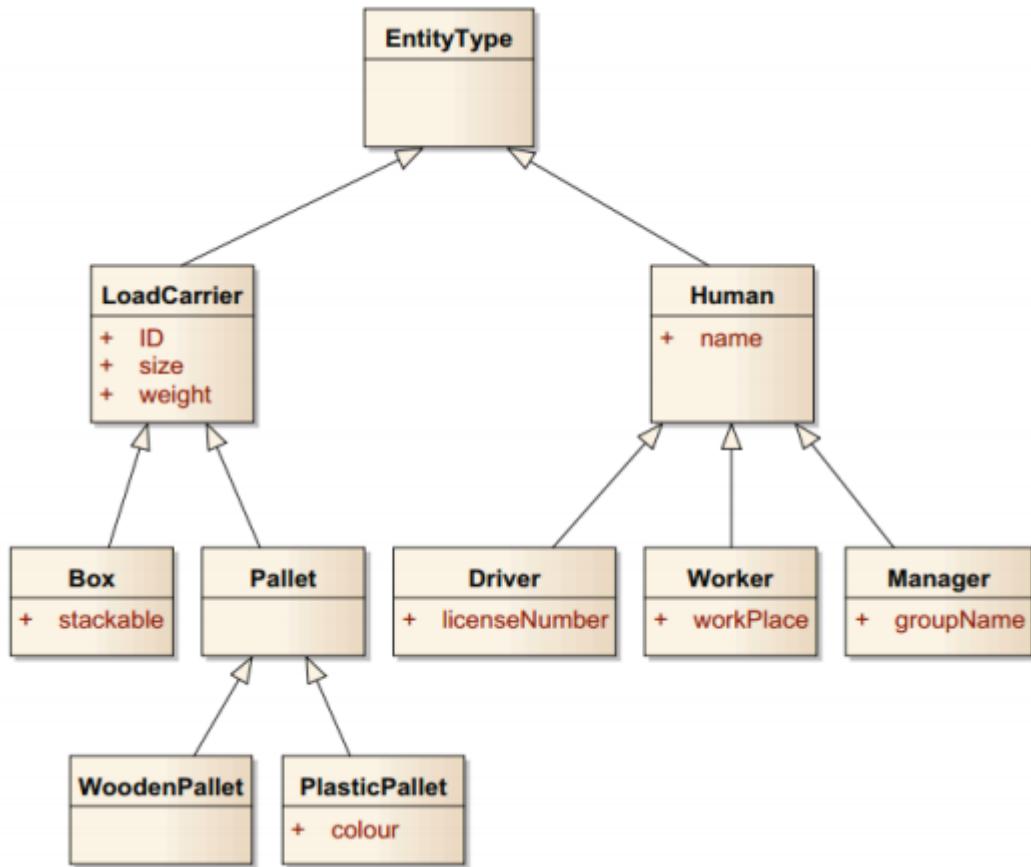


Figura 3 – Exemplo de um modelo hierárquico de *EntityType*. Fonte: Pires et al. (2015).

(DROPBOX, 2018), iCloud (ICLOUD, 2018), Google Music (GOOGLEMUSIC, 2018), Google Docs (GOOGLEDOCS, 2018), OneDrive (ONEDRIVE, 2018), entre outros.

Os termos *cloud computing* e computação nas nuvens são relativamente recentes, mas a ideia não é necessariamente nova. Alguns serviços de e-mail, como Gmail (GMAIL, 2019) e Yahoo Mail! (YAHOO, 2019); "discos virtuais" na Internet, como Dropbox (DROPBOX, 2018) ou OneDrive (ONEDRIVE, 2018); sites de armazenamento e compartilhamento de fotos ou vídeos, como Flickr (FLICKR, 2019) e YouTube (YOUTUBE, 2019), são exemplos de recursos que, de certa forma, fazem parte do conceito de computação em nuvens.

Todos os serviços mencionados não são executados no computador do usuário, mas este pode acessá-los de qualquer lugar, muitas vezes sem pagar licenças de software. No máximo, paga-se um valor pelo uso do serviço ou pela contratação de recursos adicionais, como maior capacidade de armazenamento de dados, por exemplo.

Alguns exemplos de serviços que incorporam claramente o conceito de *Cloud Computing*:

- ❑ **Google Apps ® (GOOGLEAPPS, 2018)** : pacote de serviços que o Google oferece que conta com aplicativos de edição de texto, planilhas e apresentações, ferramenta de agenda, comunicador instantâneo integrado, e-mail com o domínio próprio (por exemplo, contato@sitedoigor.net), entre outros.

- **Amazon ® (AMAZON, 2018a)**: a Amazon é um dos maiores serviços de comércio eletrônico do mundo. Para dar suporte no volume de vendas no período de Natal, a empresa montou uma gigantesca estrutura de processamento e armazenamento de dados que acabava ficando ociosa na maior parte do ano. Foi a partir daí que a companhia teve a ideia de "alugar" esses recursos, iniciativa que resultou em serviços como Solução de Armazenamento Simples (*Simple Storage Solution - S3*) para armazenamento de dados e Nuvem de Computação Elástica (*Elastic Compute Cloud - EC2*) para uso de máquinas virtuais;
- **Netflix ® (NETFLIX, 2018)**: serviço que dá acesso a filmes, seriados e documentários a partir de um pequeno valor por mês. Não é necessário efetuar download das produções, tudo é feito por *streaming*. Além disso, o usuário pode assistir cada item do acervo quantas vezes quiser e, caso interrompa a reprodução do vídeo, pode continuar mais tarde de onde parou;
- **Aprex ® (APREX, 2018)**: oferece um conjunto de ferramentas para uso profissional, como calendário, gerenciador de contatos, lista de tarefas, armazenamento de arquivos, blog, serviço de e-mail marketing, apresentações, entre outros. Tudo é feito pela Web e, no caso de empresas, é possível até mesmo inserir logotipo e alterar o padrão de cores das páginas;
- **Evernote ® (EVERNOTE, 2018)**: serviço para criação e armazenamento de notas e informações variadas que funciona como um abrangente banco de dados. Inclui ferramentas para compartilhamento, edição, organização e localização de dados. Há opções de contas gratuitas e pagas.

2.5 Sistemas Distribuídos

Os sistemas distribuídos estão em todo o lugar, ou melhor, podem ser acessados a partir de qualquer lugar. Segundo Tanenbaum e Steen (2007), um sistema distribuído é um grupo de computadores independentes entre si e até diferentes, ligados através de uma rede de dados, que se apresentam aos utilizadores como um sistema único e coerente. É uma coleção de computadores autônomos interconectados por uma rede, com software projetado para produzir uma aplicação integrada.

Utiliza-se sistemas distribuídos o tempo todo, em especial quando acessa-se recursos na Internet onde a maior parte do conteúdo é distribuído inclusive de forma geográfica. Os sistemas distribuídos são muitos utilizados também no meio empresarial, onde sistemas de informática, a muito tempo, são oferecidos de forma distribuída e dificilmente centralizado, pois a distribuição de recursos elimina parte do risco que se tem ao centralizar sistemas, principalmente quando críticos.

Para conseguir uma flexibilidade em sistemas distribuídos abertos, é necessário que o sistema seja organizado com um conjunto de componentes relativamente pequenos e de fácil substituição ou adaptação (TANENBAUM; STEEN, 2007).

No mundo da tecnologia atual, são vários os exemplos de sistemas distribuídos. Alguns que se destacam:

- ❑ Sistemas de pesquisas (motores de busca);
- ❑ Sistemas financeiros;
- ❑ Jogos Online;
- ❑ Redes Sociais e plataformas idênticas;
- ❑ Datas Centers;

2.6 UML

Com intenção de obter softwares de qualidade, existem os processos de desenvolvimentos de softwares que dividem o desenvolvimento em fases: Levantamento de Requisitos, Análise, Codificação e Testes. Para se fazer um projeto do software é necessário uma descrição de como a aplicação será implementada. Após isso, tem-se a modelagem do sistema, que é feita utilizando ferramentas de modelagens, como por exemplo, a UML (*Unified Modeling Language* - Linguagem de Modelagem Unificada) quando se refere a sistemas orientados a objetos.

Através da UML pode-se: documentar, visualizar e especificar formas gráficas para um modelo de software. A UML modela aspectos estáticos e dinâmicos de um sistema. Na versão 2.0 da UML, existem 13 tipos de diagramas divididos em 3 categorias: Estrutural, Dinâmica e Física.

- ❑ Diagramas Estruturais: O objetivo é a descrição estática de um sistema, como classes, atributos e operações, além de relacionamento entre eles. A Tabela 1 lista os diversos diagramas que pertencem a esta classificação. Os diagramas estruturais contém: Diagrama de Classes, Diagrama de Objetos, Diagramas de Caso de Uso, Diagrama de Estrutura Interna, Diagrama de Colaboração, Diagrama de Componentes.
- ❑ Diagramas Comportamentais (Dinâmica): Explica o funcionamento de parte de um sistema ou processo de negocio relacionado a tal explicação. Na Tabela 2 são apresentados os diversos diagramas que se enquadram nesta categoria. O diagrama comportamentais contém: Máquina de Estados, Diagrama de Atividades, Diagrama de Sequência, Diagrama de Tempo, Diagrama de Atividades, Diagrama de Sequência e Diagrama de Interação.

- Diagramas de Interação (Física): Diagrama que é considerado um subgrupo dos diagramas comportamentais, porque é normalmente utilizados na representação de interações entre objetos de uma aplicação. Os diferentes diagramas que fazem parte deste conjunto de representações foram descritos na Tabela 3. O diagrama de interação contém: Diagrama de Implantação e Diagrama de Pacote.

Tabela 1 – Diagramas Estruturais da UML. Modificado de (GROFFE, 2013).

Diagramas	Função
Diagrama de Classes	Permite a visualização de um grupo de classes, detalhando atributos e operações (métodos) presentes nesta última, assim como possíveis relacionamentos entre essas estruturas. Contudo esse tipo de representação também pode estar incluindo definições de interfaces.
Diagrama de Componentes	Exibe diversos componentes de um sistema, além de possíveis dependências entre tais elementos.
Diagrama de Pacotes	Detalha as dependências entre diferentes <i>namespaces</i> /pacotes que compõem uma aplicação. Dentro da plataforma .NET, um <i>namespaces</i> costuma conter classes, interfaces e outros elementos, atuando como uma forma de agrupamento lógico destes elementos.
Diagrama de Objetos	Apresenta o estado de instâncias de objetos dentro de um sistema, levando em conta para isto um intervalo de tempo específico.
Diagrama de Estrutura Composta	Usado para demonstrar a estrutura interna de uma classe, incluindo referências que apontam para outras partes de um sistema.
Diagrama de Instalação	Feito para demonstrar a estrutura de hardware adotada para a implantação de uma aplicação em um ambiente. Pode envolver dispositivos como servidores de aplicação, servidores de banco de dados, terminais de usuários etc.
Diagrama de Perfil	Possibilita a definição de novos elementos UML, permitindo assim estender os diagramas existentes com a inclusão de estruturas customizadas para uma determinada necessidade.

Tabela 2 – Diagramas Comportamentais da UML. Modificado de (GROFFE, 2013).

Diagramas	Função
Diagrama de Casos de Uso	Voltado à apresentação de funcionalidades e características de um sistema, assim como de que forma tais elementos se relacionam com usuários e entidades externas envolvidas num determinado processo.
Diagrama de Atividades	Contempla as diversas tarefas desempenhadas na execução de uma atividade, sendo utilizado geralmente na representação de processos dentro de uma empresa/organização.
Diagrama de Transição de Estados	Detalha os diferentes estados pelos quais pode passar um objeto, tomando por base a execução de um processo dentro do sistema que se está considerando.

Tabela 3 – Diagramas de Interação da UML. Modificado de (GROFFE, 2013).

Diagramas	Função
Diagrama de Sequência	Demonstra as interações entre diferentes objetos na execução de uma operação, destacando ainda a ordem em que tais ações acontecem num intervalo de tempo. A sequência em que as diversas operações são executadas ocorre na vertical, de cima para baixo.
Diagrama de Interatividade	Espécie de representação híbrida, com uma estrutura similar à de diagramas de atividade. O que diferencia este tipo de representação está justamente no fato do equivalente a uma atividade ser representada por outro diagrama, sendo o de sequência um exemplo de uso válido neste último caso.
Diagrama de Colaboração ou Comunicações	Similar aos diagramas de sequência, é também empregado na modelagem de interações entre vários objetos dentro de um determinado contexto. Este tipo de representação difere de um diagrama de sequência por não possuir uma estrutura rígida para demonstrar a comunicação entre objetos, ou seja, estes elementos podem ser dispostos na melhor ordem que se julgar necessária, sem a obrigatoriedade de exibir as diferentes operações na vertical uma após a outra.
Diagrama de Tempo	Corresponde a um tipo específico de diagrama de sequência, descrevendo mudanças de estado e interações entre objetos dentro de intervalos de tempo tomados como parâmetro.

A Linguagem Unificada de Modelagem (UML) possui diagramas conforme ilustrado na Figura 4 (representações gráficas do modelo parcial de um sistema) que são usados em combinação, com a finalidade de obter todas as visões e aspectos do sistema.

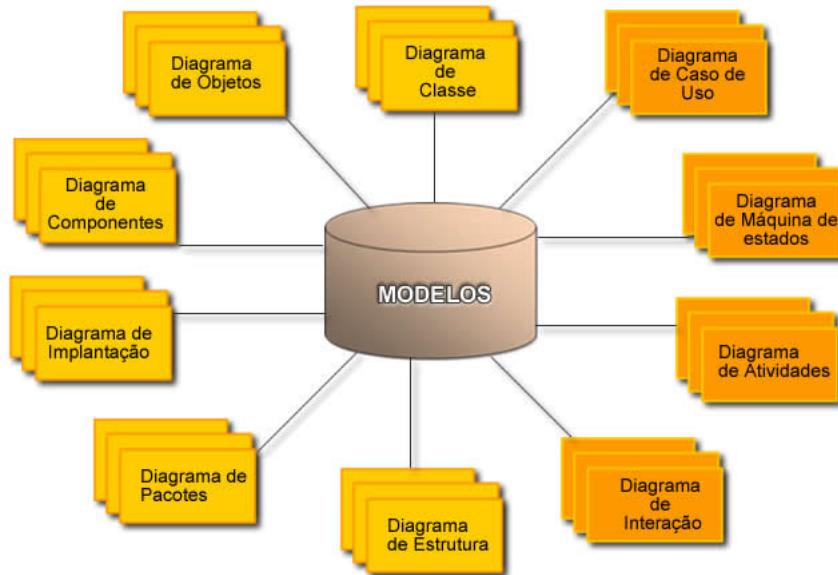


Figura 4 – Parte dos Diagramas da UML 2.0. Fonte: Martinez (2007)

A UML é uma linguagem que pode ser utilizada para estabelecer uma linguagem de modelagem visual comum, que facilitará e auxiliará o desenho e a comunicação entre objetos e utiliza-se, também, de um conjunto de técnicas de notações gráficas para a criação de sistemas. E tem sido usada de maneira efetiva em domínios como:

- ❑ Sistemas de Informações Corporativos;
- ❑ Serviços Bancários e Financeiros;
- ❑ Telecomunicações;
- ❑ Transportes;
- ❑ Vendas de Varejo;
- ❑ Eletrônicas Médica;
- ❑ Sistemas Legados;
- ❑ Sistemas Distribuídos;
- ❑ Sistemas WEB;
- ❑ Sistemas IoT;
- ❑ Sistemas *Mobile*;

Segundo Booch, Rumbaugh e Jacobson (2006), a UML não está apenas restrita para modelagem de software, pois pode ser utilizada para modelar sistemas que não sejam de software, por exemplo, um fluxo de trabalho no sistema legado, uma estrutura e comportamento de sistema de saúde ou projeto de hardware.

Para que a UML não se tornasse muito complexa, seus criadores deixaram de lado alguns detalhes, mas criaram mecanismos de extensão que a torna adaptável para sistemas com características específicas (ERIKSSON et al., 2003).

A UML provê, também, maneiras de estender a linguagem:

- ❑ Adicionando estereótipos;
- ❑ *Tagged values*;
- ❑ Restrições;

A UML sem extensões não oferece os recursos necessários para modelar todos os tipo de aplicações atuais, por exemplo, aplicações WEB, *mobile*, Sistemas Distribuídos, IoT, Sistemas Legados, dentre outras. Com isso necessitou-se criar extensões e, em meados anos 90, Jim Conallen criou uma extensão UML para aplicações WEB, denominada Extensão de Aplicativo para Web (WAE). Essa extensão permitiu utilizar elementos para representar uma aplicação WEB, tais como: página cliente, página servidor, formulários, *frames*, *links*, entre outros. A Figura 5 ilustra os estereótipos criados por Conallen (2003) para a modelagem de aplicações WEB.

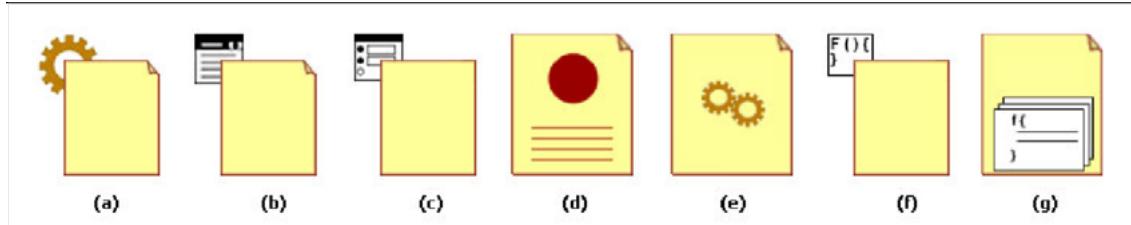


Figura 5 – Ícones dos estereótipos: (a) «server page», (b) «client page», (c) «HTML Form», (d) «static page», (e) «dynamic page», (f) «script library» de classe e (g) «script library» de componente. Fonte: Oliveira e Vahldick (2010).

Qualquer desenvolvedor pode utilizar UML para ajudar no levantamento de requisitos na sua fase inicial do software, sabendo que tem artifício para a modelagem estrutural e visual que farão parte do mesmo. A UML pode ser utilizado para esboçar estruturas de um sistema, através de desenhos de um componente ou processo do software, para melhor entendimento. O mesmo serve, também, para documentar e anexar as atividades de codificação que foi implementada. A UML é de grande valia para levar a implantação de um bom software.

Devido à semelhança entre o desenvolvimento WEB e o desenvolvimento móvel, muitas dessas extensões podem ser facilmente utilizadas ou adaptadas quando se pretende modelar aplicações que envolvem mobilidade.

CAPÍTULO 3

Experimentos e Análise dos Resultados

Nesse capítulo serão ilustrados os experimentos desenvolvidos e análises alcançadas no trabalho. Inicialmente serão apresentados os diagramas da aplicações citadas no Capítulo anterior. Para a criação dos diagramas foram utilizado os softwares: Star UML¹ e Astah UML².

3.1 Experimentos

Para exemplificar as aplicações apresentadas no Capítulo anterior, desenvolveu-se a modelagem em um cenário específico (um cadastro de seguro de veículos) apresentando o cadastro como uma maneira mais abrangente dos diagramas. As entidades representadas são: Pessoa, Pessoa Física, Pessoa Jurídica e Clientes.

O primeiro diagrama denominado Diagrama de Classe, conforme Figura 6, que permite a visualização de um grupo de classes, detalhando e exibindo os atributos e métodos em cada classe. No exemplo apresentado tem 4 classes: Pessoa, Tipo-Pessoa, Pessoa Jurídica e Física, com alguns atributos dentro dele, sendo que com esse diagrama consegue-se visualizar quais serão as variáveis do sistema.

Diagrama de Classe é um dos mais utilizados e um dos mais importantes da UML, já que ele serve de apoio para a maioria dos demais diagramas. Como o próprio nome já diz, define a estrutura das classes utilizadas pelo sistemas, e determina-se os atributos e métodos que se tem em cada classe, e também o relacionamento que estabelece entre as classes para trocar informações entre em si.

A Figura 7 ilustra o Diagrama Caso de Uso, na qual existem 2 atores, Funcionário e Cliente. Dentro desse diagrama encontram-se alguns casos de uso para demonstrar a funcionalidade do sistema: o Funcionário faz o cadastro da Pessoa, após verifica-se qual é o tipo de pessoa, podendo ser Física ou Jurídica, em seguida escolhe o tipo. Esse diagrama é o mais geral e informal da UML, utilizado normalmente nas fases de levantamento e

¹ <<http://staruml.io>>

² <<http://astah.net/editions/uml-new>>

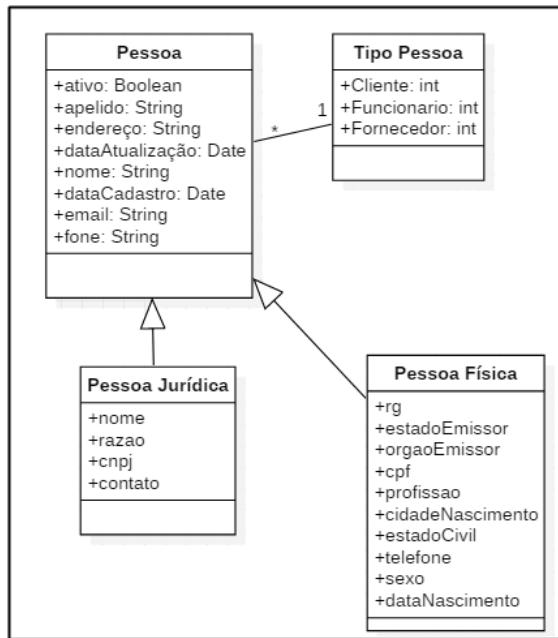


Figura 6 – Diagrama de Classe

análise de requisitos do sistema, apesar de ser consultado durante todo o seu processo de modelagem, uma vez que pode servir de base para outros diagramas. Apresenta uma linguagem simples e de fácil entendimento para que os usuários possam ter uma ideia geral de como o sistema se comportará, procurando detectar os atores (usuários, outros sistemas ou até mesmo algum hardware especial) que utilizarão de alguma forma o software, do mesmo modo que os serviços, ou seja, as funcionalidades que o sistema disponibilizará aos atores.

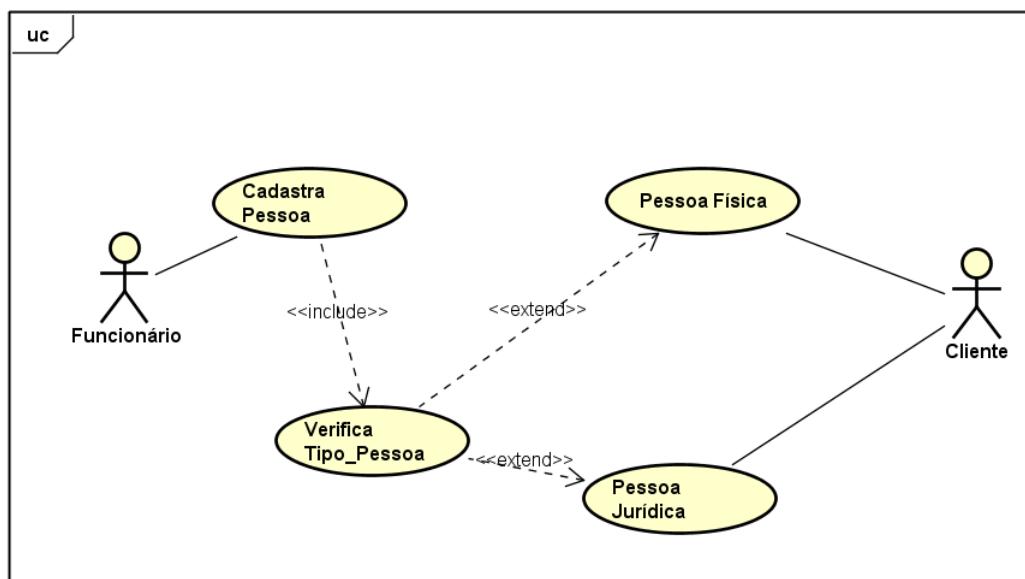


Figura 7 – Diagrama Caso de Uso

O Diagrama de Objetos está associado ao Diagrama de Classes, uma vez que o Diagrama de Objetos é praticamente um complemento do Diagrama de Classes e também muito dependente dele. Esse diagrama nos fornece uma visão dos valores armazenados pelos objetos de um Diagrama de Classe em um determinado momento da execução de um processo do software. Pode-se ressaltar que um objeto não apresentam métodos, somente atributos e, então, eles armazenam os valores contidos. A Figura 8 ilustra um Diagrama de Objetos.

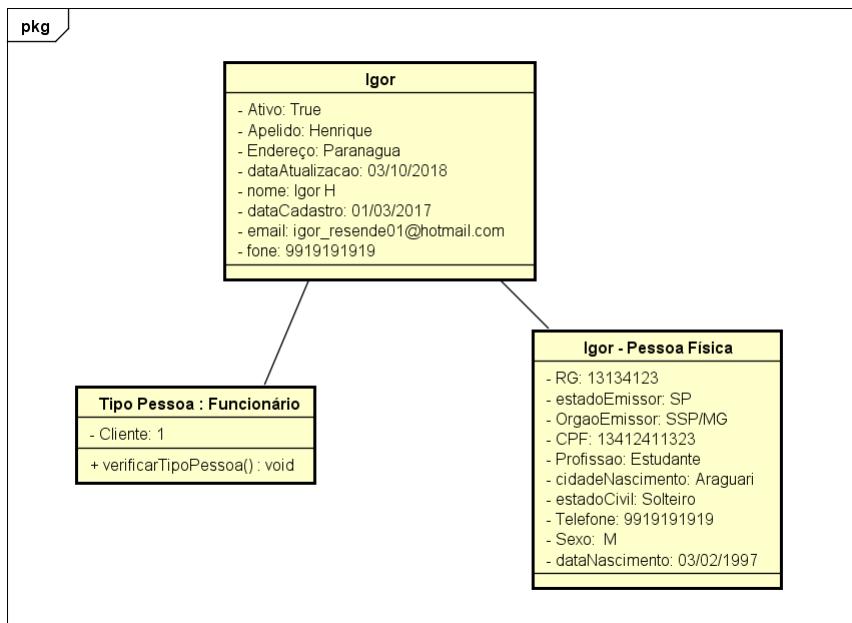


Figura 8 – Diagrama de Objetos

O Diagrama de Componentes, como o próprio nome já diz, ele identifica os componentes que fazem parte do sistema, um subsistema ou mesmo os componentes ou classes internas de um componente individual. Um componente pode ser representado tanto para componente lógico ou físico. Esse diagrama é interessante para ser utilizado como uma forma de documentar como estão estruturados os arquivos físicos de um sistema, permitindo assim, uma melhor compreensão além da reutilização de código. No exemplo (Figura 9) tem-se três componentes: um componente banco de dados e 2 componentes específicos que são páginas WEB.

O Diagrama de Pacotes descreve como os elementos do modelo estão organizados em pacotes e demonstra as dependências entre eles. Esse diagrama é vantajoso para a modelagem de subsistemas e para modelagem de subdivisões da arquitetura de uma linguagem. Pode ser aplicado para representação de conjuntos de sistemas integrados, representados por pacotes, ou ainda, os submódulos englobados por um sistema. Na Figura 10 tem-se a dependência de um controle de interface para o cadastro.

No Diagrama de Atividades (Figura 11) enfatiza-se a sequência e condições para coordenar o comportamento de baixo nível do sistema. Diagrama de Atividades também pode

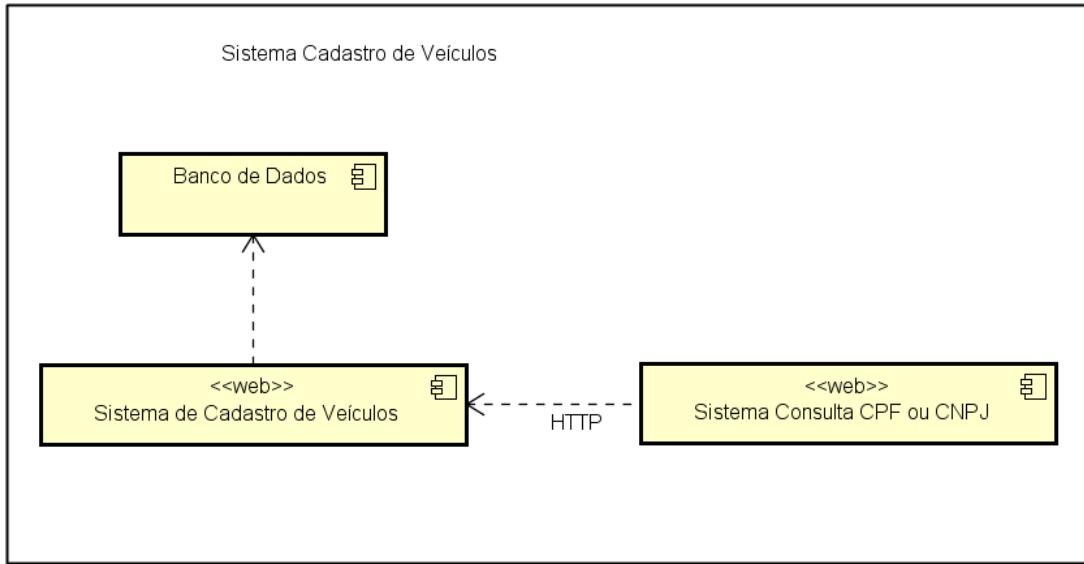


Figura 9 – Diagrama de Componentes

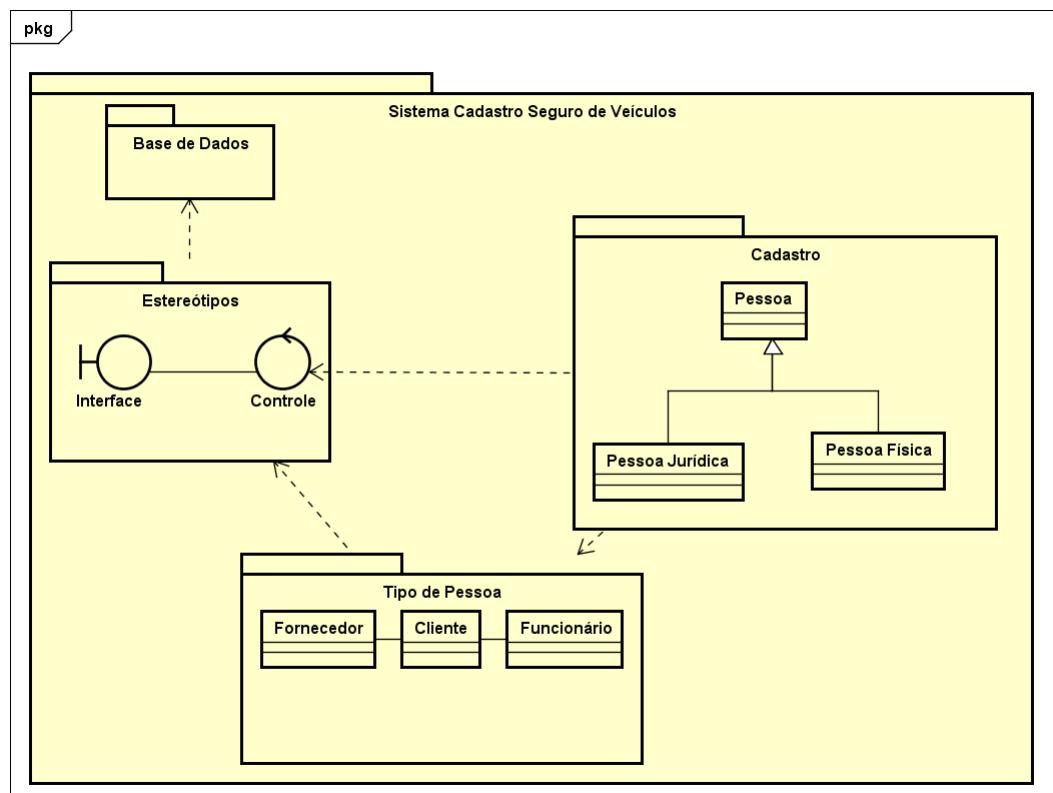


Figura 10 – Diagrama de Pacotes

ser considerado um diagrama mais detalhistas, apresentando muitas semelhanças com os antigos fluxogramas utilizados para desenvolver a lógica de programação e determinar o fluxo de controle do sistema.

No Diagrama de Sequência apresentado na Figura 12 tem-se um diagrama comportamental, ou seja, que procura determinar a sequência de eventos que ocorrem no sistema,

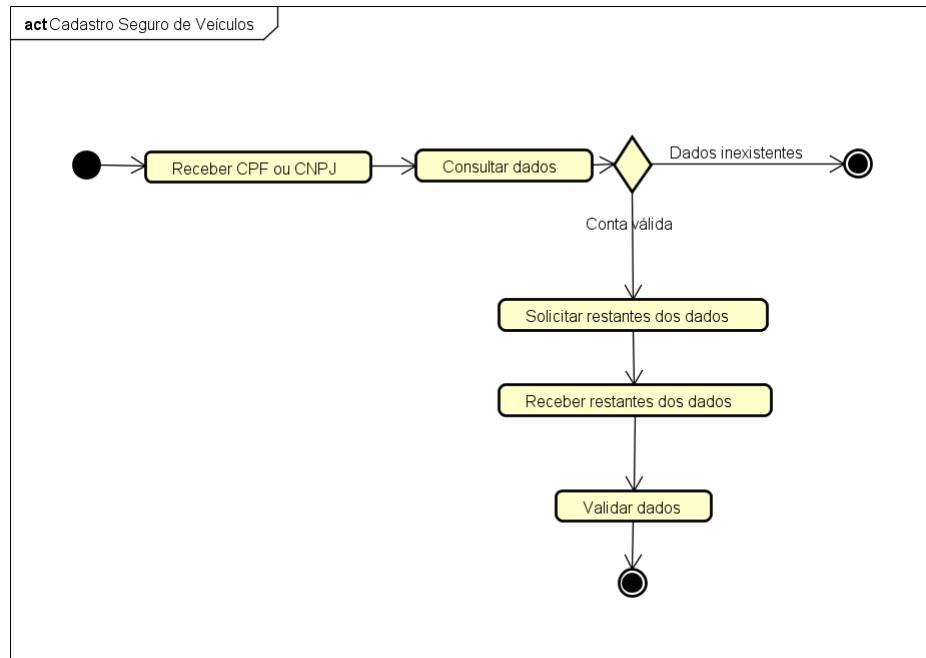


Figura 11 – Diagrama de Atividades

identificando quais mensagens devem ser disparadas entre os elementos envolvidos e em que ordem. Assim, o objetivo principal desse diagrama é determinar a ordem em que os eventos ocorrem, as mensagens que são enviadas, os métodos que são chamados e como os objetos interagem dentro de um determinado processo.

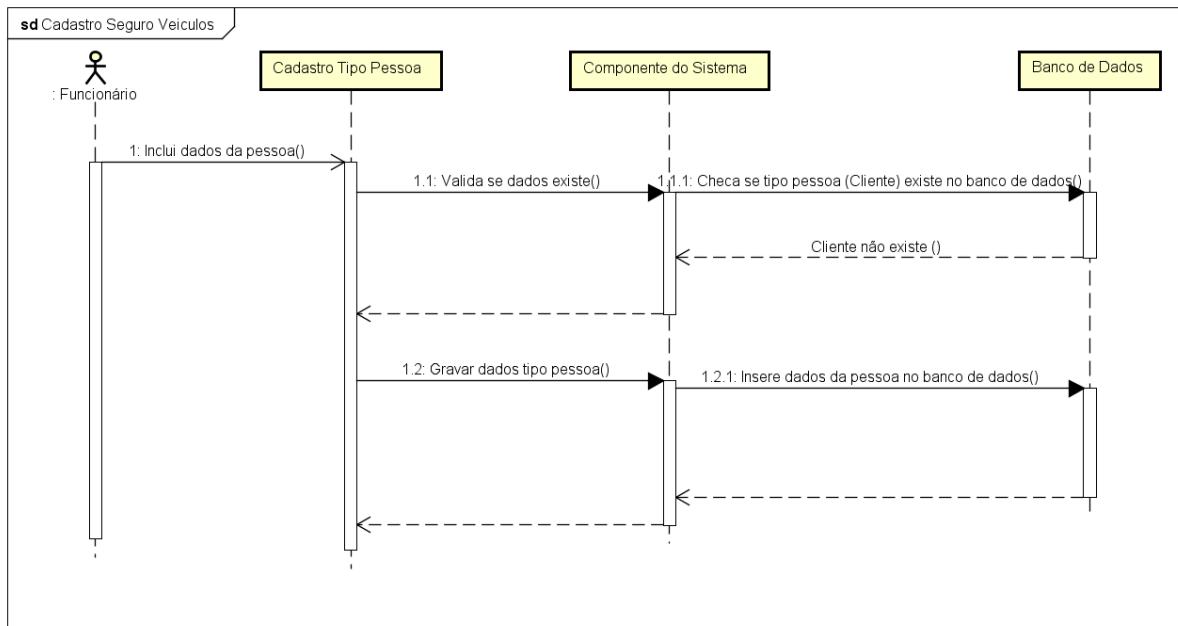


Figura 12 – Diagrama de Sequências

O Diagrama de Implementação apresenta uma visão mais física da UML. Seu enfoque é a questão da organização da arquitetura física sobre a qual o software será implantado

e executado em termos de hardware, ou seja, as máquinas (computadores pessoais, servidores, etc) que darão suporte ao sistema, além de definir como essas máquinas estarão conectadas e por meio de quais protocolos se comunicarão e transmitirão informações. Esse diagrama é mais eficiente e útil quando o sistema modelado for executado sobre múltiplas máquinas. No exemplo (Figura 13) pode-se observar que o servidor de *firewall* está associado ao servidor de comunicação. As associações, além de determinarem uma ligação física entre os nós, também podem determinar a forma que se comunicam, ou seja, o protocolo de comunicação utilizado, por meio de estereótipos. Um exemplo de nós associados utilizando estereótipos é o item «TCP/IP».

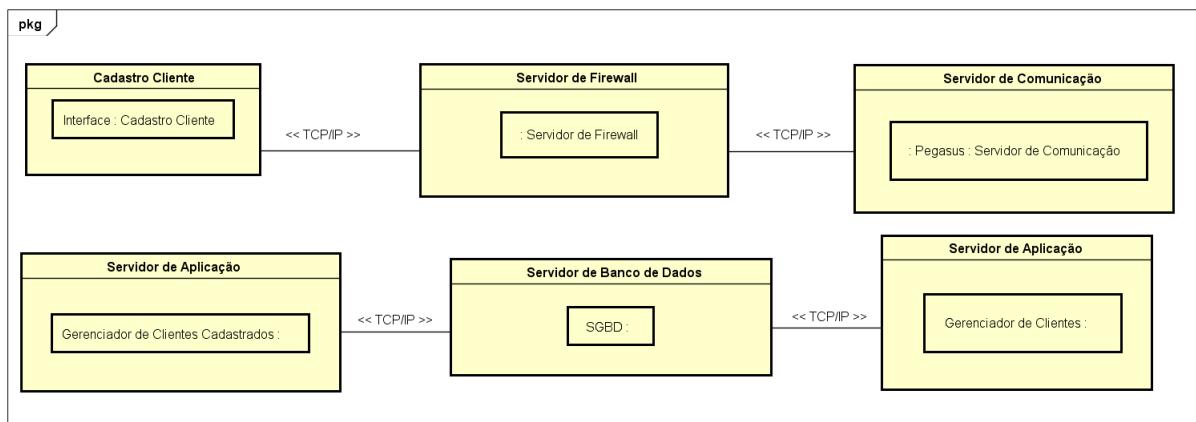


Figura 13 – Diagrama de Implantação

O Diagrama de Estado ilustrado na Figura 14 demonstra o comportamento de um elemento por meio de um conjunto finito de transições de estado, ou seja, uma máquina de estados. O diagrama além de poder ser utilizado para expressar o comportamento de uma parte do sistema, também pode ser usado para expressar o protocolo de uso de parte de um sistema.

Diagrama de Comunicação, anteriormente chamado Diagrama de Colaboração (até a versão 1.5 da UML), está amplamente associado ao Diagrama de Sequência, sendo que um completa o outro. As informações que são exibidas no Diagrama de Comunicação são, com frequência, praticamente as mesmas apresentadas no de Sequência, porém com um foco diferente, já que esse diagrama não se preocupa com a temporalidade do processos, concentrando-se em como os elementos do diagrama estão vinculados e quais as mensagens trocam entre si durante um processo. Por ser similar ao de Sequência, o Diagrama de Comunicação utiliza-se de muitos de seus componentes, como atores e objetos, incluindo também o seus estereótipos de fronteira e controle. No entanto, os objetos no Diagrama de Comunicação não têm linhas de vida (Figura 15).

O Diagrama de Estrutura Composta é um dos três novos diagramas que passaram a existir na UML 2.0. Utiliza-se para modelar colaborações. Uma colaboração descreve uma visão de um conjunto de entidades cooperativas interpretadas por instâncias que cooperam

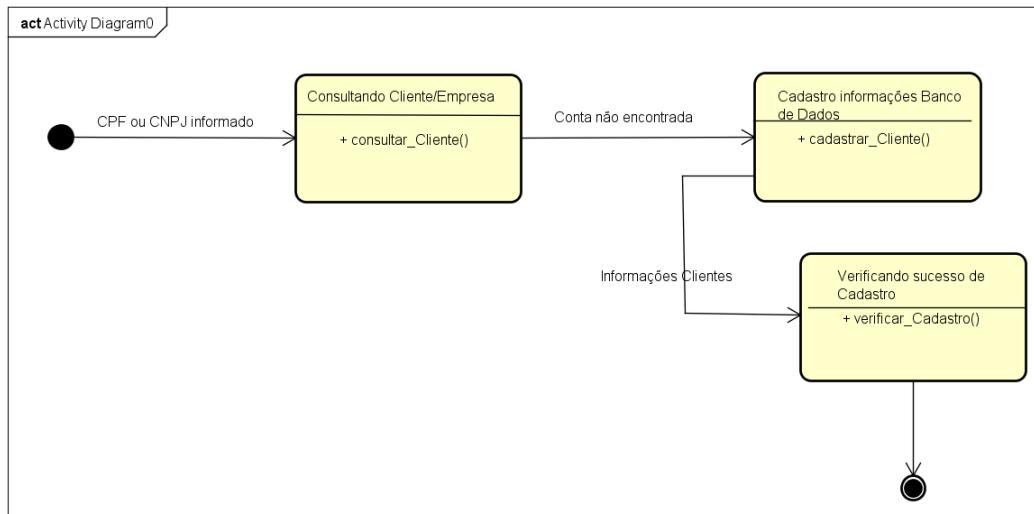


Figura 14 – Diagrama de Estado

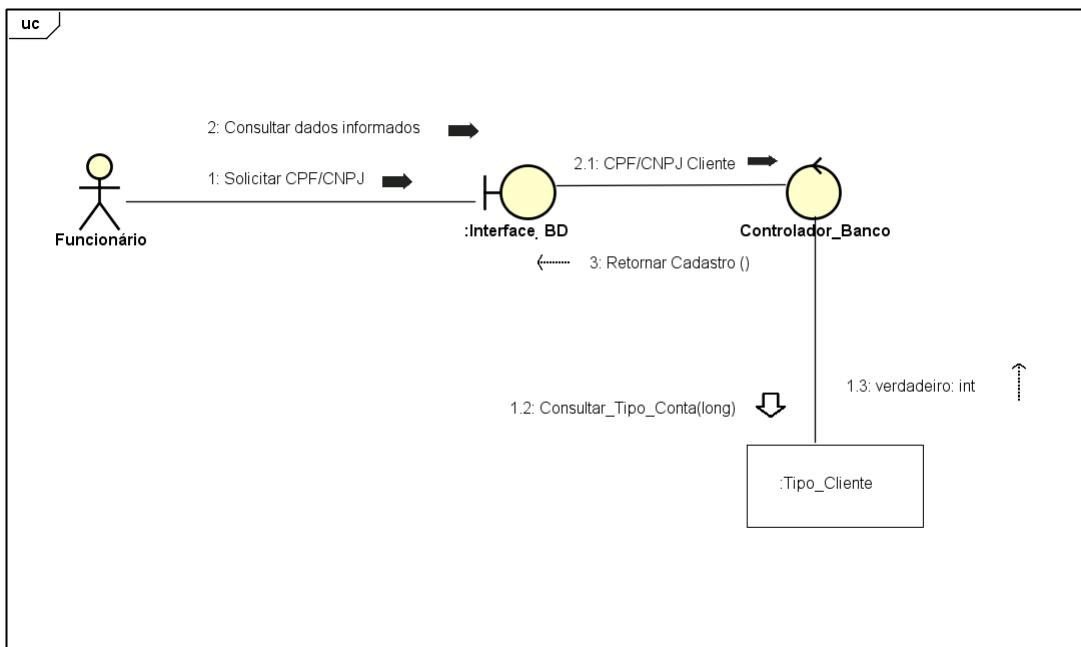


Figura 15 – Diagrama de Comunicação

entre si para executar uma função específica. O Diagrama de Estrutura Composta é semelhante ao Diagrama de Classes, porém esse último apresenta uma visão estática da estrutura de classes, enquanto o primeiro tenta expressar arquiteturas de tempo de execução. Um exemplo desse diagrama está na Figura 16.

O Diagrama de Interatividade (Figura 17) é uma variação do Diagrama de Atividade. Seu objetivo é fornecer uma visão geral do controle de fluxo. O mesmo utiliza-se quadros no lugar dos nós de ação, embora símbolos como o nó de decisão e o nó inicial, sejam também utilizados.

A UML é flexível e possibilita adicionar extensões a ela, possibilitando então modela-

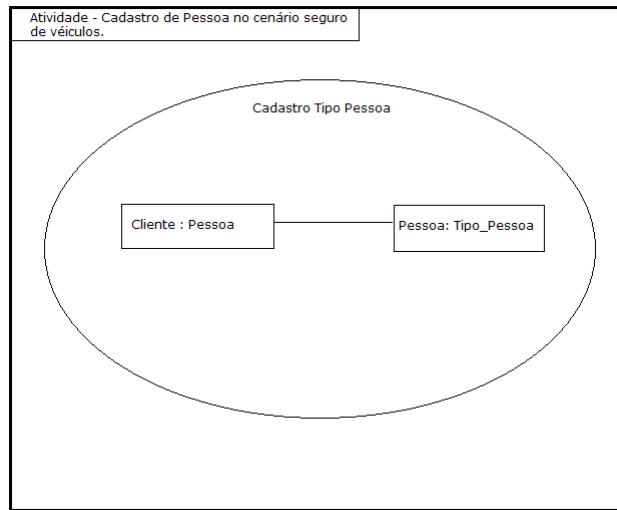


Figura 16 – Diagrama de Estrutura Composta

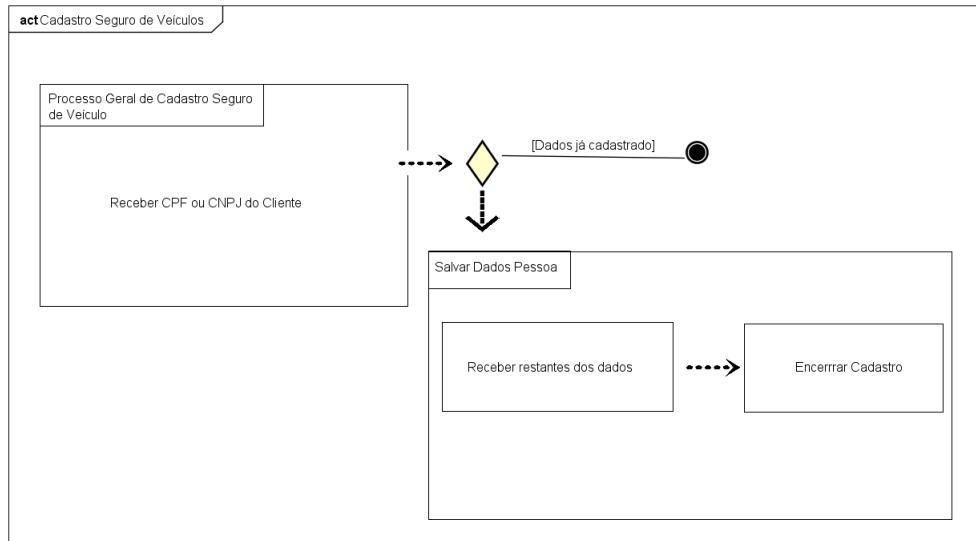


Figura 17 – Diagrama de Interatividade

gens de aplicações com características específicas. Encaixam-se nesse perfil as aplicações WEB. Sem uma técnica que permita especificar um modelo de aplicação WEB com suas particularidades é provável que sua integridade e rastreabilidade fiquem prejudicadas. Com o uso da extensão WAE, ilustrada na Figura 18, é possível demonstrar os elementos que tem importância do ponto de vista da modelagem.

Para a representação de uma aplicação em IoT, não foi possível encontrar extensões ou diagramas em específicos para atender o mesmo, com isto, a representação foi feita através do Diagrama de Caso de Uso. A Figura 19 ilustra os softwares das aplicações com suas funcionalidades e as ligações que se tem em IoT.

No Diagrama *Cloud* após várias tentativas em diversos diagramas, o diagrama que encaixou-se para representação e uma visibilidade mais detalhada foi o Diagrama de

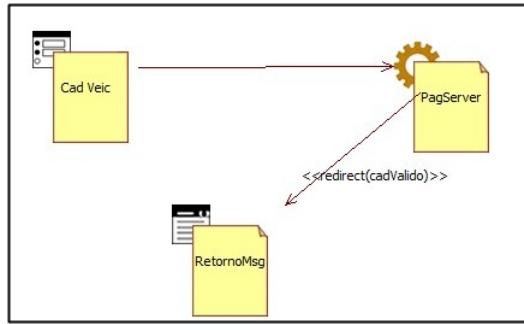


Figura 18 – Diagrama Web - WAE

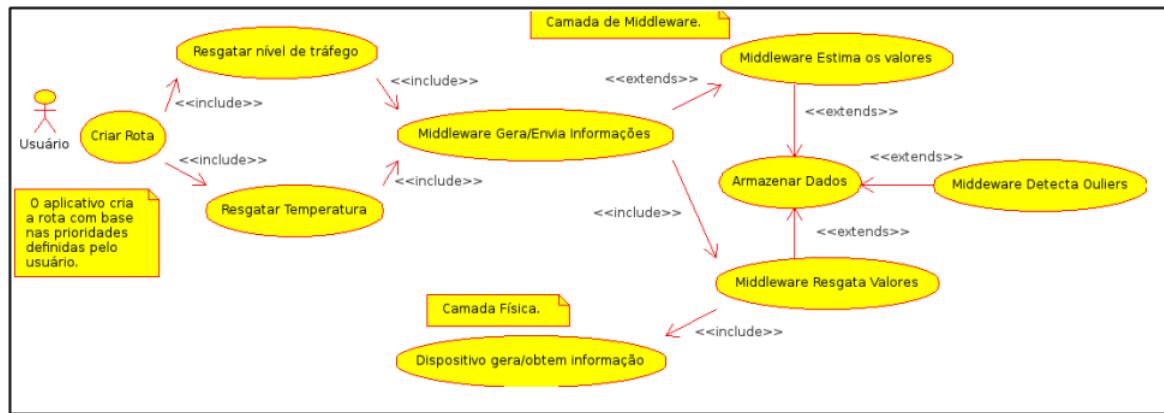


Figura 19 – Diagrama IoT

Deployment, que é um Diagrama Estrutural (especifica a estrutura do software) em que descreve os componentes de hardware e software e sua interação com outros elementos de suporte ao processamento. Representa a configuração e a arquitetura de um sistema em que estarão ligados seus componentes, sendo representado pela arquitetura física de hardware (Figura 20).

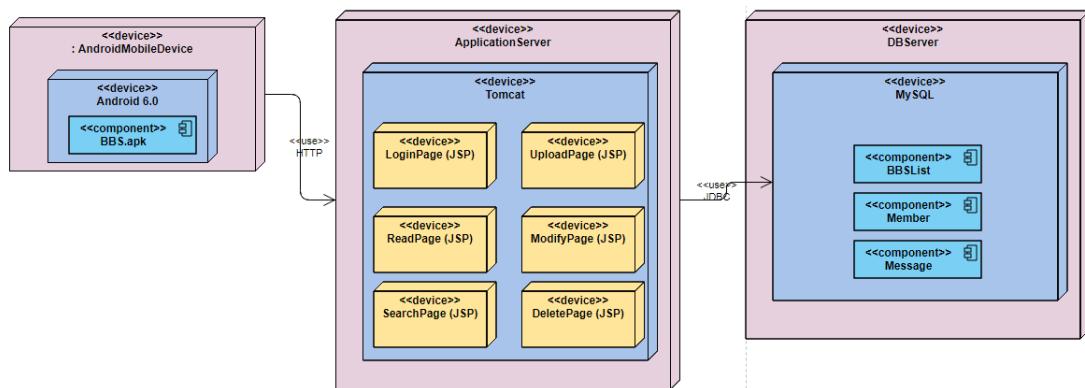


Figura 20 – Diagrama Cloud

3.1.1 Modelagem de um Sistema Moderno

Conforme demonstrado na seção 3.1, para certos tipos de aplicações não existem diagramas específicos, tornando obrigatório a utilização de algum que pudesse representar a ideia a ser modelada no sistema.

O sistema Moderno é “complexo” a ser modelado, como estudo de caso, chama-se “Sala de Aula Inteligente com IoT”. Esse projeto foi parcialmente desenvolvido e implementado em (SILVA; RIBEIRO, 2018; SILVA, 2019), porém ocorreram problemas ao se fazer a modelagem do sistema, apesar dos diagramas terem sido utilizados como Diagramas de Atividades e Sequências para demonstrar a ideia geral do sistema. Essa dificuldade serviu como um grande incentivo para o desenvolvimento deste trabalho.

O problema resume-se em utilizar os dispositivos dos próprios usuários (*smartphones* e/ou *tablets*) para ter a interação entre o professor, os alunos e o material didático. Em uma ideia geral, o projeto utiliza um sistema cliente/servidor conectado a uma rede LAN e Internet, disponibilizando para os discentes e docentes por meio de *wireless*, o acesso à página de login e o material didático necessário para a aula que será ministrada, além da projeção e distribuição do material. A Figura 21 demonstra, de forma resumida, a proposta do trabalho a ser modelado.

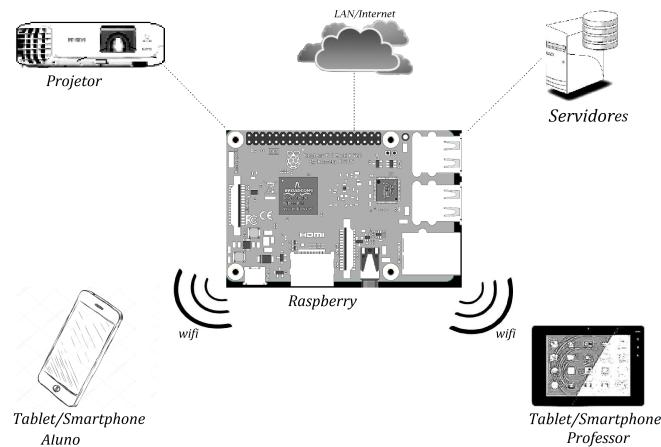


Figura 21 – Ideia geral do projeto a ser modelado. Fonte: (SILVA, 2019).

Explicando de forma sucinta, têm-se um Raspberry Pi (IoT) que será o responsável por grande parte da aplicação. Este fornecerá opção de login, pois têm-se a integração de dados com um SGBD, um servidor que armazena a quantidade de faltas, e outro para armazenamento e reconhecimento de imagens dos alunos. O Raspberry Pi comunica com os dispositivos conectados (professor e os alunos) realizando e processando ações realizadas pelos mesmos. Existe mais uma forma de interação que se dá por meio dos repositórios locais, em nuvens e armazenamentos externos. Finalizando, a interação ocorre por meio da projeção pela saída HDMI para o projetor multimídia. O Raspberry Pi possui acesso a rede LAN e acesso a WEB. Concluindo, o Raspberry Pi tem o papel de gerenciar, realizar leitura de hardwares e processar os dados solicitados para os devidos encaminhamentos.

Para obter uma visibilidade melhor do sistema proposto utilizou-se o Diagrama de Sequência, representando todas aplicações em um único sistemas. No Diagrama de Sequência estão separados por Atores e em diagramas distintos. O primeiro caso é a representação do Login Aluno (Figura 22), demonstrando como o Aluno faz o login do sistema e, então o Raspberry busca essas informações digitadas pelo aluno. Caso os dados sejam válidos, é feito o reconhecimento facial do aluno para a gerar uma confirmação de presença na aula do dia. Para isso, o Raspberry Pi busca essa imagem em Servidor contendo as imagens dos alunos (IMG_SERVER). Após verificado a imagem e, se estiver correta a identificação, a presença do aluno é salva no banco de dados (SGBD).

Feito isso, o Professor disponibilizará o material da aula do dia. Então o Raspberry Pi seleciona esse material no Servidor e inicia a projeção da aula e, também, envia a projeção para o celular do aluno. Ao final da aula, o aluno preenche uma avaliação da aula ministrada pelo professor, logo após essa avaliação é armazenada no banco de dados (SGBD).

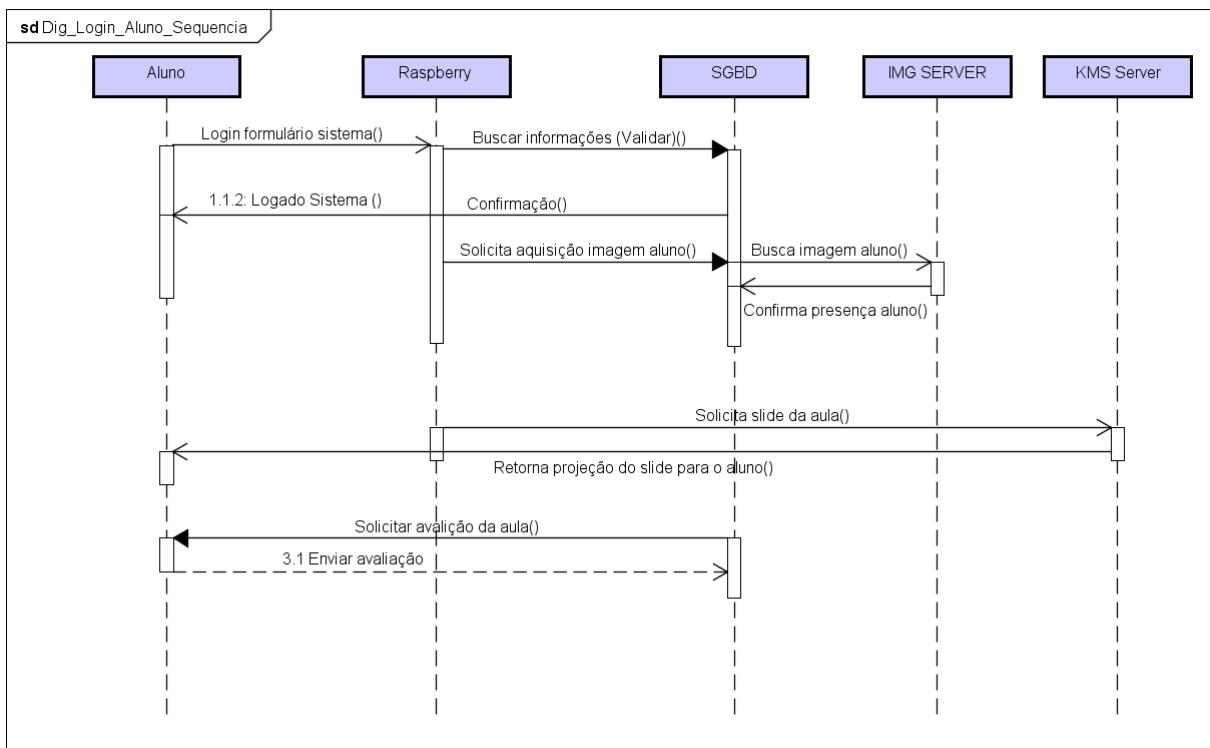


Figura 22 – Diagrama Sequência - Login Aluno

Para a representação do lado do Professor, é necessário representar em três diagramas, pois o professor pode ter armazenado os materiais em repositórios diferente e também, em armazenamento externo. A Figura 23 ilustra o processo de sequência: o professor faz o login no sistema, então o Raspberry Pi valida essa informação no Banco de Dados. Caso esteja correto, será exibida a lista de disciplinas ministradas pelo professor, no qual todo processamento é feito no IoT com a busca feita no MOODLE, após retornada a disciplina,

o Professor seleciona o material e o exibe no projetor. Esse material é posteriormente enviado ao dispositivo do aluno. Ao final da aula é exibida a lista de presença dos alunos, para que o professor confirme as presenças dos alunos assinaladas pelo sistema.

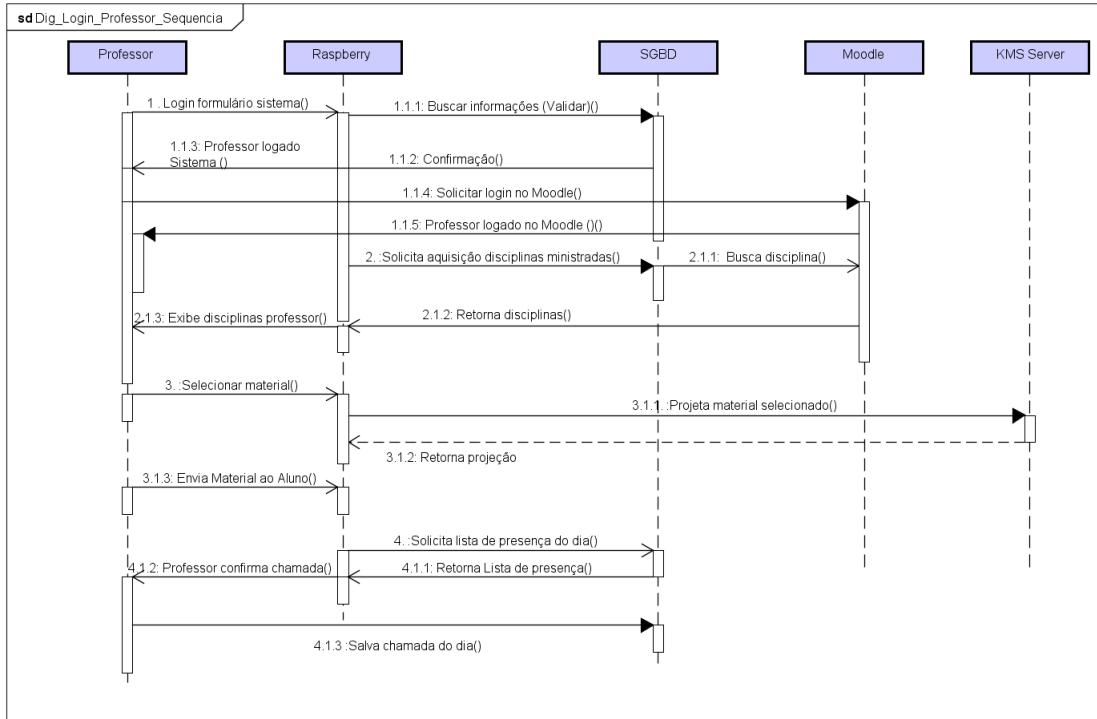


Figura 23 – Diagrama Sequência (Moodle) - Login Professor

Os outros Diagramas de Sequência do Professor têm-se o mesmo raciocínio dos demais, sendo que a Figura 24 está representando a mesma lógica do primeiro diagrama de sequência, no qual o processamento será no material das pastas da USB do Professor, fazendo as outras ações lógicas iguais.

Por último, a Figura 25 representa a ligação do Login Professor com seus materiais e disciplinas armazenados na nuvem (*CLOUD*), visto que o professor pode optar ter os materiais em nuvem para ter um acesso e segurança dos seus arquivos.

3.1.2 Modelando com Casos de Uso

O Diagrama de Casos de Uso, apresenta os elementos (hardwares, aplicações) externos e as funcionalidades de um sistema.

Seguindo a representação com os Diagrama de Casos de Uso, tem-se na Figura 26 o Ator "Aluno" com suas funcionalidades. Após logar-se no sistema, receberá a projeção da aula em seu dispositivo, e também receberá o material da aula que será salvo localmente. Ao final da aula será necessário avaliar a aula ministrada pelo Professor.

Na Figura 27 é representado o Diagrama Caso de Uso do Raspberry Pi (IoT) que é definido como o ator principal do sistema, responsável por processar, responder requisি-

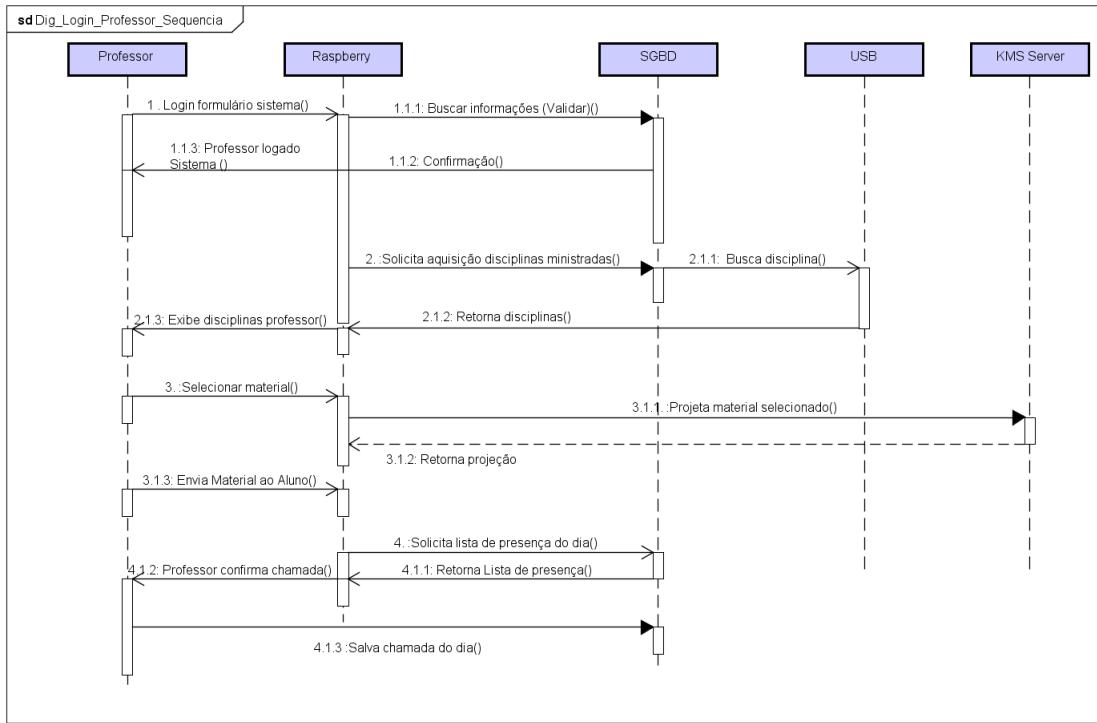


Figura 24 – Diagrama Sequência (USB) - Login Professor

ções, dos demais dispositivos, grande parte das funcionalidades do sistemas é necessário passar por ele, pois ele é o intermediário para responder e enviar requisições feita pelos servidores, alunos e professor.

Na representação do Caso de Uso do Ator Professor, na Figura 28, poderá seguir dois caminhos diferentes: ao logar no sistema, terá a possibilidade de fazer o login no Moodle Acadêmico escolhendo a disciplina, ou então pegar o material em um repositório. Feita a escolha, o IoT processará o arquivo escolhido e enviará para os alunos. Ao final da aula, o professor será responsável por confirmar a presença dos alunos.

É possível representar os dispositivos (Figura 29) demonstrando suas funcionalidades principais, e interações com os materiais.

Nas Figuras 30 e 31 nota-se que os atores são responsáveis pelo armazenamentos de dados, armazenamento dos materiais, armazenamento das imagens dos alunos e armazenamento das frequências dos discentes.

3.1.3 Extensões UML para Aplicações Web

Com as restrições que se pode notar para modelar sistemas mais complexos com os diagramas disponibilizados pela UML, concluiu-se que para a melhor representação do sistema proposto a utilização da extensão WAE, pois permite a integração com estereótipos e artefatos primários de aplicações, o lado servidor e lado cliente. Essa extensão foi criada, após o problema de representar sistemas web quando uma página possui *scripts*

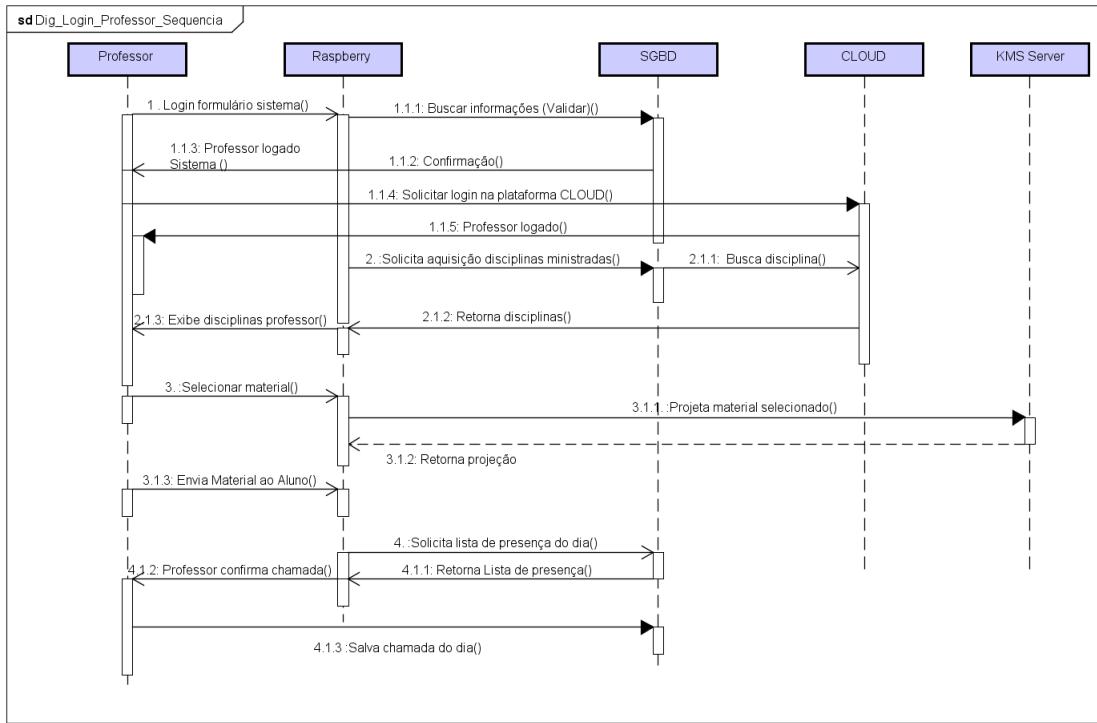


Figura 25 – Diagrama Sequência (CLOUD) - Login Professor

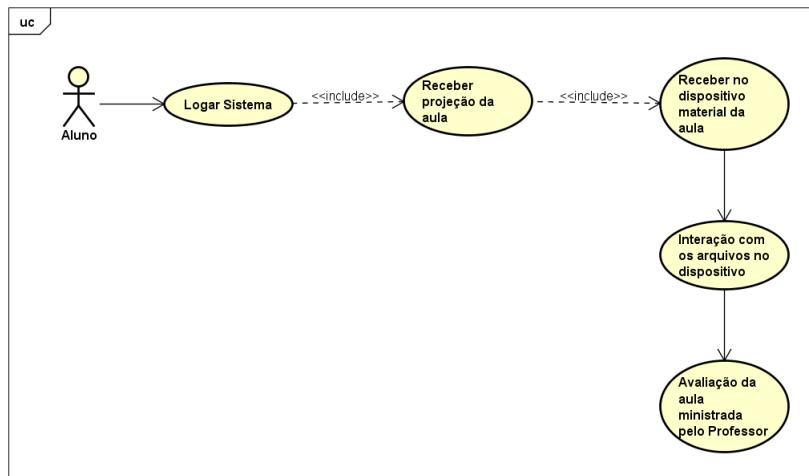


Figura 26 – Diagrama Caso de Uso - Aluno

a serem executados no servidor e parte no cliente. A notação e diagramas da UML não é suficiente, quando o assunto é demonstrar a ligação de *scripts* como objetos na páginas web, com isso surgiu a necessidade de modificar a UML.

A Extensão de Aplicativo para Web (WAE), proposta por Conallen (2003), é uma extensão da notação UML com semânticas adicionais e restrições, a fim de permitir a modelagem arquitetural de elementos web específicos. Esta extensão da UML permite a captura, desenvolvimento e a análise de como será a execução das regras de negócio nas páginas web, objetivando modelar páginas web, relacionamentos entre as mesmas,

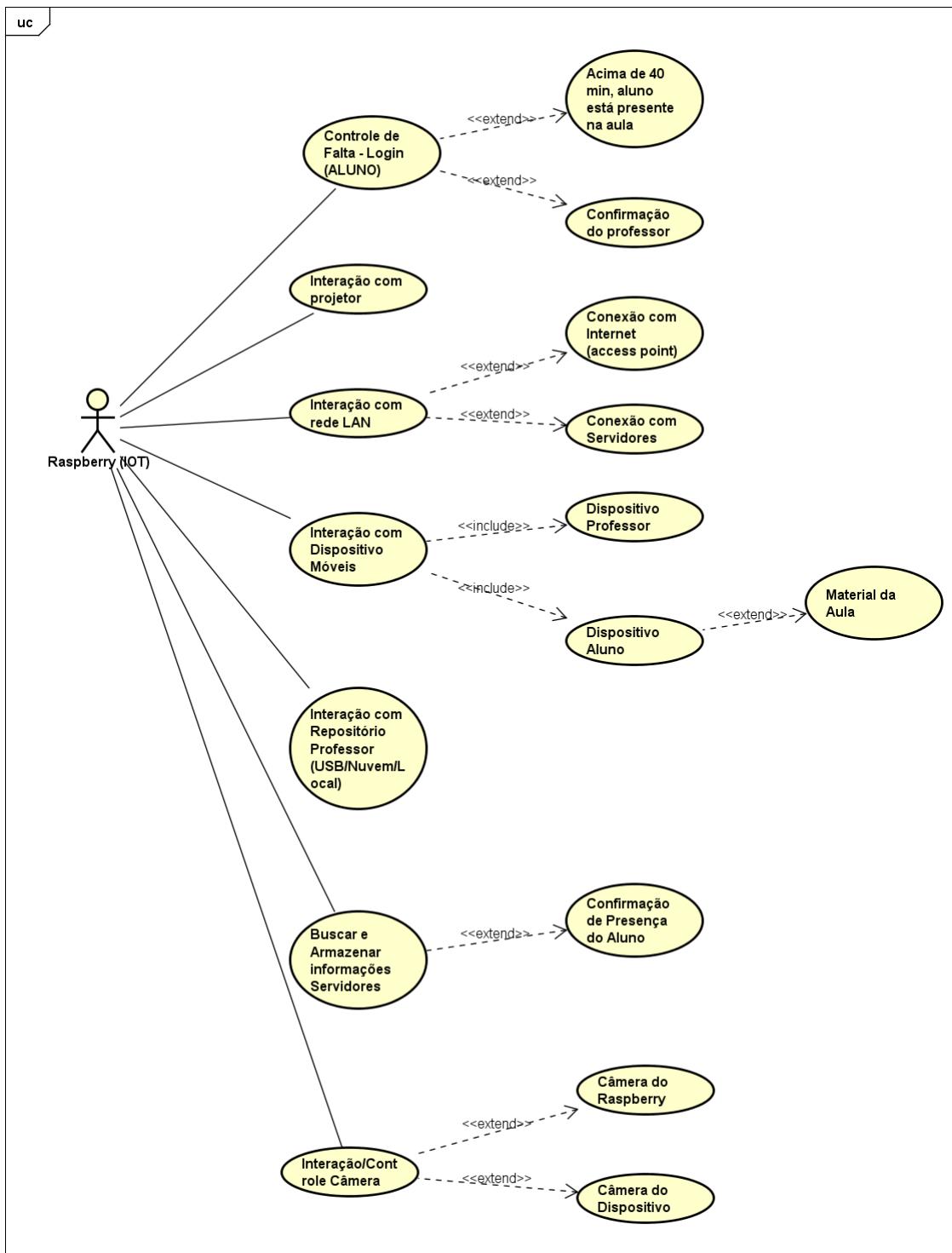


Figura 27 – Diagrama Caso de Uso - IoT

rotas de navegação, *scripts* no lado do cliente e geração de páginas no lado do servidor, permitindo o relacionamento entre os elementos específicos da web e os demais elementos do sistema.

Os elementos da WAE são: estereótipos de páginas, relacionamentos entre páginas, *forms* e *frameset*.

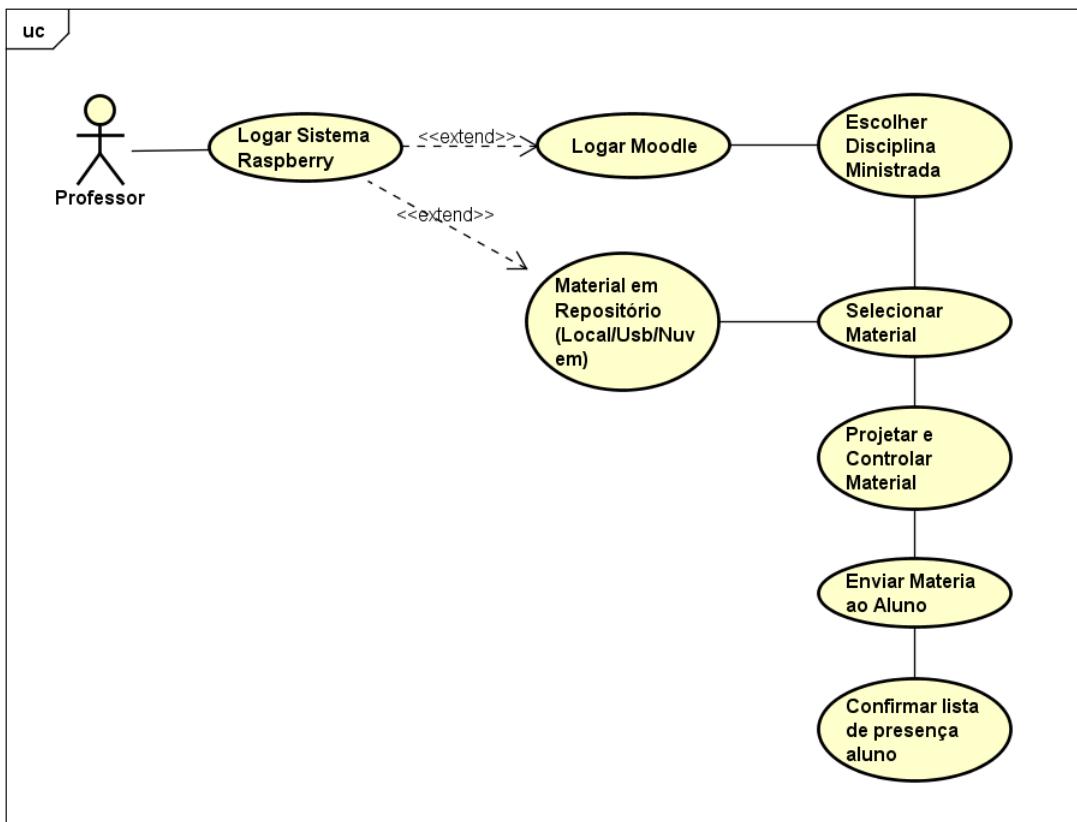


Figura 28 – Diagrama Caso de Uso - Professor

Estereótipo de Páginas

Páginas que já existem em aplicações Web apresentam características distintas de acordo com o local onde seus métodos são executados. Uma página que roda no servidor é completamente diferente de páginas que rodam em *browsers* de clientes, possuindo acesso a informações que não são acessíveis pelo cliente, sendo controladas de maneira distinta.

A extensão WAE representa essas páginas no modelo como duas classes diferentes: Página Servidor «server page» e Página Cliente «client page» (Figura 32). A página servidor relaciona-se com componentes que existem no servidor e serão utilizados nessas páginas. Já o cliente relaciona-se com componentes que existem no clientes (*browsers*).

Relacionamento entre as páginas

Existe uma ligação de relacionamento entre páginas clientes e servidores, visando à comunicação entre essas classes. Esse relacionamento auxilia a definir o mapa do site. Existem diferentes mecanismos de relacionamentos entre essas classes:

- ❑ **Builds:** relacionamento unidirecional entre um «server page» e uma «client page» indicando que uma página servidor geralmente é responsável pela construção de páginas clientes. Representado pelo estereótipo «build»;

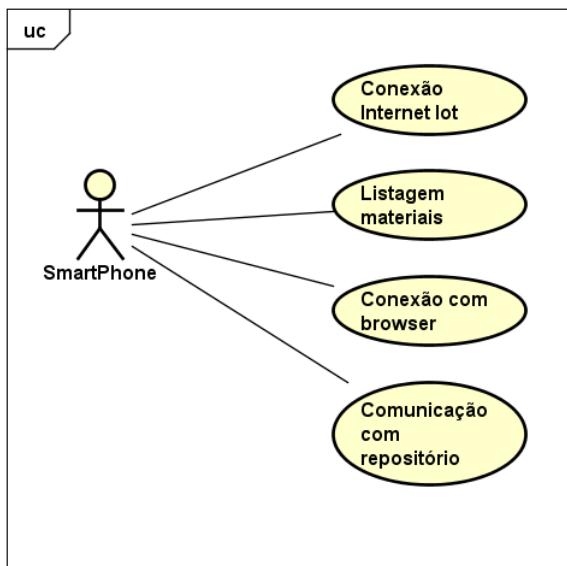


Figura 29 – Diagrama Caso de Uso - SmartPhone

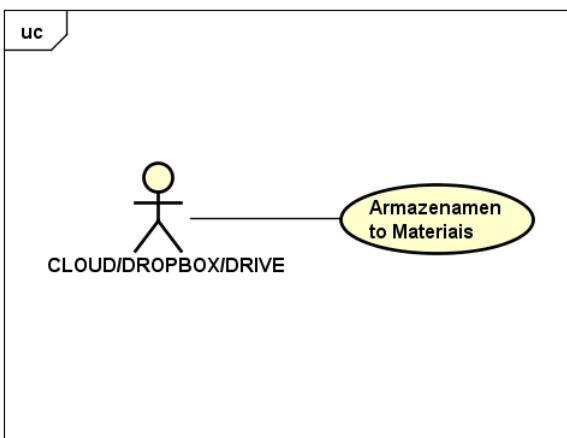


Figura 30 – Diagrama Caso de Uso - Cloud

- **Links:** um relacionamento adicional e fundamental no projeto de aplicações Web é *hyperlink*. Páginas Clientes possuem *hyperlinks* para outras páginas Web. Esse estereótipo «links» é definido para associação entre páginas clientes e outras páginas.

Forms (Formulários)

São expressos no modelo como uma agregação de páginas clientes. Um objeto «form» existe somente no contexto cliente. O mesmo aceita entradas do usuário e submete a página servidor para processar. Em um formulário pode haver diversos *forms*, cada um possuindo uma ação diferente na página. Um estereótipo de associação «submit» representa o relacionamento entre um formulário e a página WEB que faz o processamento.

Frameset

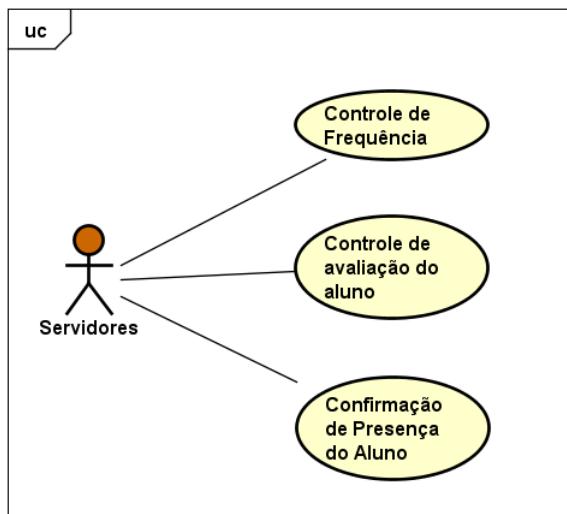


Figura 31 – Diagrama Caso de Uso - Servidores

Permite que o projetista Web possa dividir a janela do *browser* em sub-áreas retangulares, cada uma diferente páginas, ilustrado na Figura 32. Para coordenar atividades entre páginas em *frames* requer habilidade para referenciar páginas contendo *frames*. Target é o termo usado para quando uma página cliente referencia outra página Web ou *frame*. O *target* não tem propriedade ou atributos, ele é simplesmente uma referência possível para uma página cliente.

Um estereótipo precisa ser definido para associação que indica que uma página cliente está requerendo um link para ser rodado em uma outra janela. Um estereótipo «targeted link» é aplicado para associação entre páginas clientes e *targets* com quem elas interagem. Parâmetros que são passados para o servidor com o «targeted link» podem ser identificados com um atributo link da UML.

As principais combinações entre estereótipos e relacionamentos válidos na WAE estão na Tabela 4.

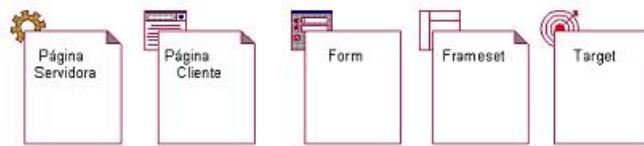


Figura 32 – Estereótipos na notação WAE

Após a breve explicação do funcionamento do WAE, a Figura 33 representa o Login do sistema proposto. Ao entrar na aplicação, uma página de boas vindas é exibida ao usuário (professor ou aluno). Essa página abrirá a Tela Login do sistema que contém um formulário para o login no sistema e, após o preenchimento deste formulário, é feito um *submit* ao servidor do banco de dados para a validação dos dados. Caso retorne valores verdadeiros então a página será redirecionada para a Tela Principal do Sistema. Pode ser

Tabela 4 – Combinação de Estereótipos com relacionamentos válidos.

De: Para:	Página Cliente	Servidor de Páginas	Conjunto de Quadros (Frameset)	Alvo (Target)	Formulário
Página Cliente	<<link>> <<redirect>>	<<link>> <<redirect>>	<<link>> <<redirect>>	Dependência	Agregação
Servidor de Páginas	<<redirect>> <<build>>	<<redirect>>	<<redirect>> <<build>>	Não tem	Não tem
Conjunto de Quadros (Frameset)	<<frame content>>	Não tem	<<frame content>>	<<frame content>>	Não tem
Alvo (Target)	Não tem	Não tem	Não tem	Não tem	Não tem
Formulário	Agregação	<<submit>>	Não tem	Não tem	Não tem

notado que junto da representação, existe uma classe com os *scripts* em JavaScript que faz parte para a validação da aplicação de logar.

Contudo, não foi possível exemplificar o sistema proposto como um todo, pois existem estereótipos que não atendem as necessidades do sistema. Por este motivo fez-se necessária a sugestão de novos estereótipos para a extensão, para se ter uma visão mais clara entre os componentes conectado, como: IoT, Cloud, Smartphone, entre outros. Na Figura 34 estão representados os estereótipos necessários para a modelagem do sistema.

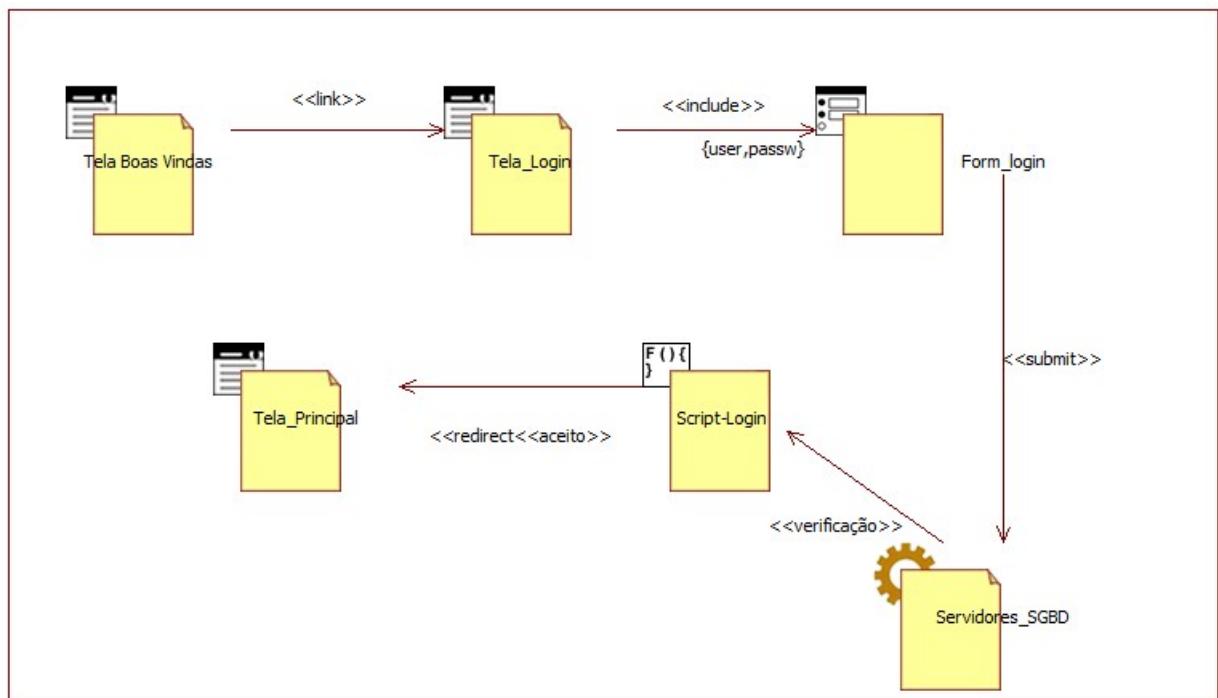


Figura 33 – Representação Login no Sistema - WAE.

Após a criação e sugestão dos estereótipos para modelagem do sistema, foi possível

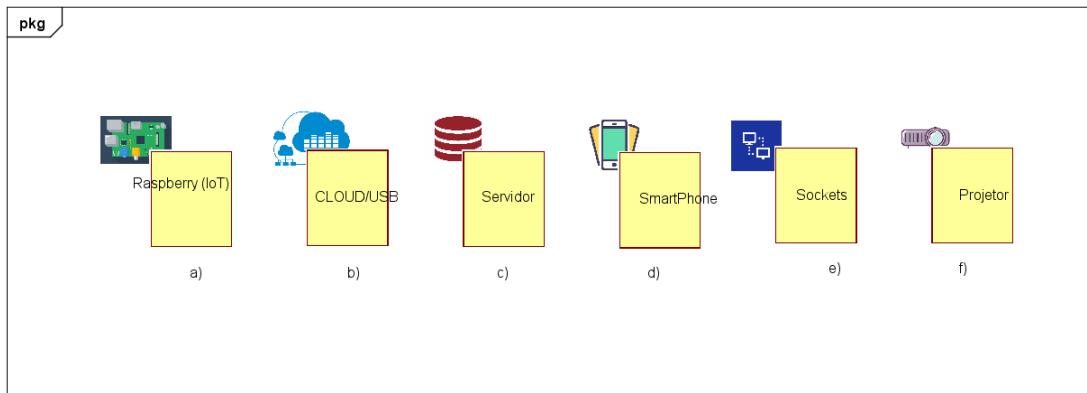


Figura 34 – Sugestões de novos ícones dos estereótipos WAE: (a) «device IoT», (b) «Cloud», (c) «Servers», (d) «device mobile» (e) «Sockets Comunicação», (f) «Projetor Multimídia».

exemplificar o sistema completo em um único diagrama (Figura 35). Poderia ter sido criado estereótipos com cores diferentes, porém em uma possível impressão para o programador não ficaria de maneira evidente para diferenciação (impressão preto/branco), tornando-se mais adequada a utilização de imagens ao fundo de cada estereótipos.

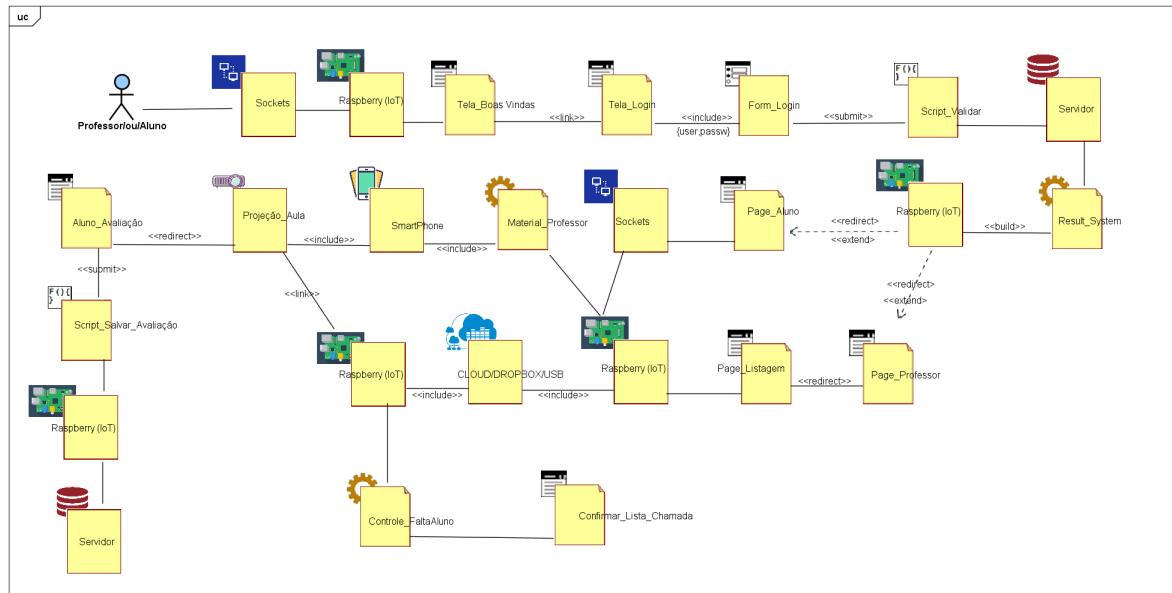


Figura 35 – Modelagem WAE Proposta - Sala de Aula Inteligente com IoT.

3.2 Avaliação dos Resultados

Após a grande dificuldade para a modelagem de um sistema moderno e "complexo", que envolve várias aplicações em um único sistema, foi possível demonstrar que ficou de maneira evidente e claro quais as ligações sequências de cada estereótipos, e também a

representação ilustrativa de cada um, com a extensão WAE facilitou-se essa representação, apesar de não ter sido eficiente totalmente com os próprios estereótipos, a UML nos permite criar classes, estereótipos, ligações para ficar de maneira mais clara para o desenvolvedor de como seria a aplicação antes da sua implementação final.

As representações de estereótipos sugeridas (Figura 34) foram necessárias para atender a modelagem do sistema proposto.

Ao fim do trabalho, foi possível diferenciar os diagramas disponibilizados pela UML, podendo gerar indicativos de qual modelagem se adapta ao sistema final do usuário, demonstrando-se formas de criações de extensões para facilitar o entendimento sistemas mais complexos.

Algumas limitações a se ressaltar no projeto foram:

- ❑ Devido aos softwares existentes não permitirem fazer algumas ligações, pois não são estereótipos do programa, com isso foi necessária a ajuda de softwares terceiros;
- ❑ Com os diagramas sem extensões, fica evidente a dificuldade para interagir várias aplicações em um único diagrama;
- ❑ Criação de estereótipos apenas através de cores, gerando uma dificuldade de diferenciar qual estereótipo pertence, em uma possível impressão;
- ❑ Dificuldade para compreender vários diagramas interligando;

CAPÍTULO 4

Conclusão

Esse trabalho apresentou uma proposta de auxiliar e gerar indicativos para a modelagem de aplicações como IoT, Cloud, Mobile, Rede Mundial de Computadores (WEB). O desenvolvimento desse trabalho foi constituído de algumas etapas principais, sendo elas: fundamentação teórica, experimentos e análise em um sistema simples com os principais diagramas e por fim exemplificação em um sistema no mundo real.

Constatou-se que ainda não havia um perfil UML direcionado para a modelagem de interfaces gráficas dentro do sistema proposto. Com isso, o tema para o trabalho foi definido e estudos mais aprofundados foram feitos, constatando que algumas das extensões para aplicações WEB, descritas na Seção 3.1.3, poderiam ser adaptadas para o desenvolvimento com várias aplicações.

A partir das informações obtidas com a fundamentação teórica, o perfil UML foi desenvolvido, com o intuito de identificar problemas relativos à usabilidade na fase de modelagem do sistema. Dessa forma, os principais elementos de uma aplicação desenvolvida para ambientes com várias aplicações foram modelados, como a criação de estereótipos para servidor, o dispositivo móvel, IoT, Sockets, Cloud/USB, entre outros, atendendo às restrições impostas pela UML.

Após a criação dos elementos necessários, foi feita a exemplificação, utilizando uma aplicação real, que foi parcialmente desenvolvida por SILVA (2019). Com isso, pode-se comprovar que o perfil se torna eficiente na modelagem de interfaces gráficas para aplicações em ambientes com várias aplicações, uma vez que as interfaces são criadas levando-se em conta as limitações impostas pelo dispositivos ligados ao IoT.

A criação de interfaces gráficas para este tipo de problema se torna complicada uma vez que vários aplicações é ligado a servidores, chamadas de sockets, interação com dispositivos de alunos, o que compromete a utilização de todos os recursos disponíveis. Outra limitação encontrada é a utilizações de softwares disponíveis na internet para criações de tais diagramas.

4.1 Principais Contribuições

No Capítulo 3 (seção 3.2) tentou-se ilustrar a diferença de usabilidade e visibilidade com as criações dos estereótipos com a extensão WAE propostas com baseado no sistema real modelado. Infelizmente, modelando-se com os diagramas fornecidos pelo UML, ficou confuso para entender quais era as funcionalidades e sequências em cada aplicação.

Para sistemas que envolvem mais de uma aplicação, é viável utilizar extensões para tornar mais clara e evidente para o usuário e o desenvolvedor como eles os diagramas e estereótipos comunicam-se.

4.2 Trabalhos Futuros

Como proposta para trabalhos futuros destacam-se:

- ❑ Implementar e modelar o trabalho com suas futuras funcionalidades.
- ❑ Avaliar o modelo por meio de outros estudos de casos de Sistemas Modernos.
- ❑ Implementar extensões em ferramentas CASE (através de *plugins*, por exemplo).
- ❑ Propor integrar determinados estereótipos em algum software de desenvolvimento, sem ter que utilizar softwares terceiros para ajudar a modelar e criar estereótipos.
- ❑ Estereótipos e *Tags* mais complexos para modelar em ambientes modernos.

4.3 Contribuições em Produção Bibliográfica

A proposta desse trabalho foi publicada e apresentada nos Anais da TECHnoMONTE 2k18, Monte Carmelo - MG (RESENDE; RIBEIRO, 2018).

Referências

- ALHARBI, M. et al. Validation of fitbit-flex as a measure of free-living physical activity in a community-based phase iii cardiac rehabilitation population. **European Journal of Preventive Cardiology**, v. 23, n. 14, p. 1476–1485, 2016. PMID: 26907794. Disponível em: <<https://doi.org/10.1177/2047487316634883>>. Citado na página 17.
- AMAZON. **Amazon**. 2018. Disponível em: <<https://aws.amazon.com/pt/ec2/>>. Citado na página 20.
- _____. **AmazonDrive**. 2018. Disponível em: <<https://www.amazon.com.br/gp/feature.html?ie=UTF8&docId=1001156211>>. Citado na página 18.
- APREX. **Aprex**. 2018. Disponível em: <<http://www.aprex.com.br>>. Citado na página 20.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. CAMPUS - RJ, 2006. ISBN 9788535217841. Disponível em: <<https://books.google.com.br/books?id=ddWqxcDKGF8C>>. Citado 2 vezes nas páginas 12 e 24.
- CARPLAY. **Carros vendidos no Brasil com Android Auto e Apple CarPlay**. 2018. Disponível em: <<https://carrosbr.com/carros-no-brasil-android-auto-apple-carplay/>>. Citado na página 17.
- CHAN, S. S. et al. Usability for mobile commerce across multiple form factors. **J. Electron. Commerce Res.**, v. 3, p. 187–199, 2002. Citado na página 16.
- CONALLEN, J. **Desenvolvendo aplicações Web com UML: tradução da segunda edição**. [S.l.]: ELSEVIER EDITORA, 2003. ISBN 9788535212099. Citado 2 vezes nas páginas 24 e 39.
- CYBIS, W. de A.; BETIOL, A.; FAUST, R. **Ergonomia e usabilidade: conhecimentos, métodos e aplicações**. Novatec, 2007. ISBN 9788575221389. Disponível em: <<https://books.google.com.br/books?id=F4kmAeDBScEC>>. Citado na página 15.
- DROPBOX. **DropBox**. 2018. Disponível em: <<https://www.dropbox.com>>. Citado na página 19.
- ERIKSSON, H.-E. et al. **UML 2 Toolkit**. [S.l.]: Wiley Publishing, 2003. ISBN 0471463612, 9780471463610. Citado na página 24.

EVERNOTE. **Evernote**. 2018. Disponível em: <<https://evernote.com/intl/pt-br>>. Citado na página 20.

FIGUEIREDO, C.; NAKAMURA, E. F.; FIGUEIREDO, S. **COMPUTAÇÃO MÓVEL: NOVAS OPORTUNIDADES E NOVOS DESAFIOS**. 2015. Disponível em: <https://www.researchgate.net/publication/268435975_Computacao_Movel_Novas_Oportunidades_e_Novos_Desafios_COMPUTACAO_MOVEL_NOVAS_OPORTUNIDADES_E_NOVOS_DESAFIOS>. Citado na página 15.

FLICKR. **Flickr**. 2019. Disponível em: <<https://www.flickr.com>>. Citado na página 19.

GMAIL. **Gmail**. 2019. Disponível em: <<https://gmail.com>>. Citado na página 19.

GOOGLEAPPS. **GoogleApps**. 2018. Disponível em: <<https://gsuite.google.com.br/intl/pt-BR/>>. Citado na página 19.

GOOGLEDOCS. **GoogleDocs**. 2018. Disponível em: <<https://www.google.com/docs/about/>>. Citado na página 19.

GOOGLEMUSIC. **GoogleMusic**. 2018. Disponível em: <<https://play.google.com/music/listen>>. Citado na página 19.

GROFFE, R. J. **Modelagem de sistemas através de UML: uma visão geral**. 2013. Disponível em: <<https://www.devmedia.com.br/modelagem-de-sistemas-atraves-de-uml-uma-visao-geral/27913>>. Citado 3 vezes nas páginas 8, 22 e 23.

ICLOUD. **iCloud**. 2018. Disponível em: <<https://www.icloud.com>>. Citado na página 19.

IEEE. **The Internet of Things: The connected Revolution**. 2014. Disponível em: <http://theinstitute.ieee.org/ns/quarterly_issues/timar14.pdf>. Citado na página 16.

_____. **Towards a definition of the Internet of Things**. 2015. Disponível em: <https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf>. Citado na página 16.

JONES, M. et al. Improving web interaction on small displays. In: **Proceedings of the Eighth International Conference on World Wide Web**. New York, NY, USA: Elsevier North-Holland, Inc., 1999. (WWW '99), p. 1129–1137. Disponível em: <<http://dl.acm.org/citation.cfm?id=313234.313014>>. Citado na página 16.

KOSCIANSKI, A.; SOARES, M. dos S. **Qualidade de Software - 2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. [S.l.]: Novatec, 2007. ISBN 9788575221129. Citado na página 13.

MARTINEZ, M. **UML**. 2007. Disponível em: <<https://www.infoescola.com/wp-content/uploads/2010/03/DiagramasdaUML.jpg>>. Citado 2 vezes nas páginas 6 e 23.

MICHAELIS, H. **MOBILE**. [S.l.]: Melhoramentos, 2006. Citado na página 15.

- NETFLIX. **Netflix**. 2018. Disponível em: <<https://www.netflix.com>>. Citado na página 20.
- OLIVEIRA, E. D.; VAHLDICK, A. Um estudo de caso de utilização da wae para uml em aplicações gwt. **Congresso Sul Brasileiro de Computação (SULCOMP)**, 1 2010. Citado 2 vezes nas páginas 6 e 25.
- OLIVEIRA, S. de. **Internet das Coisas com ESP8266, Arduino e Raspberry Pi**. NOVATEC, 2017. ISBN 9788575225813. Disponível em: <<https://books.google.com.br/books?id=E8gmDwAAQBAJ>>. Citado na página 16.
- ONEDRIVE. **OneDrive**. 2018. Disponível em: <<https://onedrive.live.com/about/pt-br/>>. Citado na página 19.
- PIRES, P. F. et al. Plataformas para a Internet das Coisas. In: MARTINELLO, M.; ROBEIRO, M. R. N.; ROCHA, A. A. de A. (Ed.). **XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – Minicursos**. Porto Alegre, RS, Brasil: SBC, 2015. p. 110–169. Citado 3 vezes nas páginas 6, 18 e 19.
- RESENDE, I. H. C.; RIBEIRO, T. P. **A UML e Extensões: modelagem de aplicativos modernos**. 2018. TECHnoMonte. Citado na página 48.
- SCULLY, P. **The Top 10 IoT Segments in 2018**. 2018. Disponível em: <<https://iot-analytics.com/top-10-iot-segments-2018-real-iot-projects/>>. Citado 2 vezes nas páginas 6 e 17.
- SILVA, C. M. **Sistema de controle de apresentação por meio de Raspberry Pi**. 2019. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) - Universidade Federal de Uberlândia, Monte Carmelo-MG, Brasil. Citado 4 vezes nas páginas 6, 13, 35 e 47.
- SILVA, C. M.; RIBEIRO, T. P. **Sala de Aula Tecnológica Baseada em IoT**. 2018. TECHnoMonte. Citado na página 35.
- SILVA, R. **Amazon anuncia Echo, uma torre-assistente virtual para ajudar em casa**. 2014. Disponível em: <<https://tecnoblog.net/169174/amazon-echo-assistente-virtual/>>. Citado na página 17.
- TANENBAUM, A.; STEEN, M. V. **Sistemas distribuídos: princípios e paradigmas**. Pearson Educación, 2007. ISBN 9788576051428. Disponível em: <<https://books.google.com.br/books?id=r2SGPgAACAAJ>>. Citado 2 vezes nas páginas 20 e 21.
- TAURION, C. **Cloud Computing: Computação em Nuvem: Transformando o mundo da tecnologia da informação**. [S.l.]: Rio de Janeiro, 2009. Citado na página 18.
- URBANHUB. **Smart City 3.0 – pergunte a Barcelona sobre a próxima geração de cidades inteligentes**. 2018. Disponível em: <<http://www.urban-hub.com/pt-br/cities/barcelona-deixa-a-cidade-ainda-mais-inteligente/>>. Citado na página 17.
- YAHOO. **Yahoo**. 2019. Disponível em: <<https://br.yahoo.com>>. Citado na página 19.

YOUTUBE. **Youtube**. 2019. Disponível em: <<https://www.youtube.com>>. Citado na página 19.