



MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
(Real Academia de Artilharia Fortificação e Desenho - 1792)
CURSO DE GRADUAÇÃO EM ENGENHARIA ELETRÔNICA

GABRIEL CESARIO BUGINGA

SISTEMA DE OBJETOS INFERÊNCIAIS (SOI)
FRAMEWORK PARA INTERNET DAS COISAS

21 DE SETEMBRO DE 2017

INSTITUTO MILITAR DE ENGENHARIA

SISTEMA DE OBJETOS INFERÊNCIAIS (SOI) FRAMEWORK PARA INTERNET DAS COISAS

O Projeto de Fim de Curso apresentado ao Curso de Graduação em Engenharia Eletrônica do Instituto Militar de Engenharia, como requisito para a obtenção do título de bacharel em Engenharia Eletrônica.

Orientador: Amarildo da Costa - D.Sc.

Brasil

21 DE SETEMBRO DE 2017

c2017

INSTITUTO MILITAR DE ENGENHARIA

Praça General Tibúrcio, 80 – Praia Vermelha

Rio de Janeiro – RJ CEP: 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmар ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

621.3	Buginga, Gabriel Cesario
B931s	Sistemas de objetos inferenciais (SOI), framework para internet das coisas / Gabriel Cesario Buginga; orientado por Amarildo da Costa – Rio de Janeiro: Instituto Militar de Engenharia, 2017.
	58p. : il.
	Projeto de Fim de Curso (PROFIC) – Instituto Militar de Engenharia, Rio de Janeiro, 2017.
	1. Curso de Engenharia Eletrônica – Projeto de Fim de Curso. 2. Internet das Coisas. 3. Prolog. I. Costa, Amarildo da. II. Título. III. Instituto Militar de Engenharia.

INSTITUTO MILITAR DE ENGENHARIA

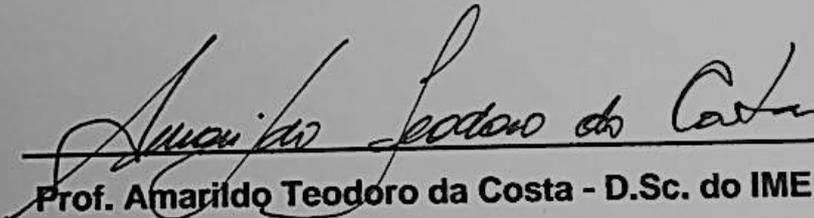
GABRIEL CESARIO BUGINGA

SISTEMA DE OBJETOS INFERÊNCIAIS (SOI) FRAMEWORK PARA INTERNET DAS COISAS

O Projeto de Fim de Curso apresentado ao Curso de Graduação em Engenharia Eletrônica do Instituto Militar de Engenharia, como requisito para a obtenção do título de bacharel em Engenharia Eletrônica.

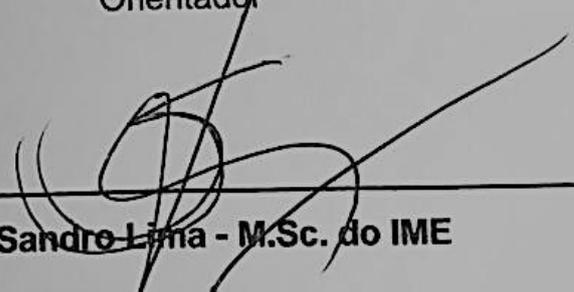
Orientador: Prof. Amarildo da Costa - D.Sc.

Aprovado em 04 de setembro de 2017 pela seguinte banca examinadora:

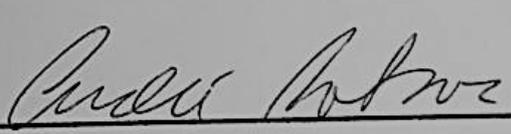


Prof. Amarildo Teodoro da Costa - D.Sc. do IME – Presidente

Orientador



Maj. Sandro Lima - M.Sc. do IME



Prof. André Rotava - M.Sc. do IME

Rio de Janeiro
2017

Resumo

Internet das Coisas tornou-se um tópico extremamente debatido e falado em diversos campos da engenharia, causando investimentos e projeções que podem chegar a bilhões em possibilidade de mercado. De automação residencial a controle remoto de paciente com doenças crônicas, sistemas sob esse paradigma parecem ubíquos e aparentemente sem uma definição no mínimo precisa.

Nesse ímpeto, o presente trabalho explorou a área de filosofia da tecnologia a fim de reavaliar a definição do que é Internet das Coisas e quais pontos de mudança qualitativa nos usos da tecnologia encontravam-se implícitos. Com isso uma nova visão de objetos sendo extensões das faculdades humanas, como a cognição ou visão, apresentou-se. Dentro dessa nova perspectiva, o entrelaçamento dos objetos pela Internet equivaleria a mesma funcionalidade de comunicação entre os órgãos do corpo humano, com uma existência digital, mas função semelhante.

Abstraindo das particularidades de cada sistema, montou-se o objetivo do trabalho como a construção de um framework geral para Internet das Coisas coerente com a filosofia delineada e que possua um poder inferencial de construção de conhecimento.

Como resultado, foi utilizada a ferramenta Prolog, como implementação de lógica de primeira ordem, na interconexão e entrelaçamento dos objetos produzindo inferências, ou seja, relações que estavam implícitas e se tornaram explícitas. Por fim para exemplificar o poder do framework foi instanciado uma aplicação básica e mínima do framework em um caso de automação residencial.

Palavras-chave: Internet das Coisas, filosofia da tecnologia, framework, lógica de primeira ordem, Prolog.

Abstract

Internet of Things became an extremely debated topic inside many engineering fields, causing investments and predictions that can reach billions of market share. From home automation to remote control of chronic diseased patients, systems under this paradigm look ubiquitous and apparently without a rigorous definition.

Within this impetus, the current work explored the field of philosophy of technology in order to re-evaluate Internet of Things' definition and which points of qualitative change on its use were implicit. Using these insights, a new vision about objects as extensions of the human faculties, like cognition or vision, presented itself. Inside these new perspective, the entanglement of the objects via the Internet amount to the same communicative functionality between body organs, with a digital existence, but similar functions.

Abstracting from each system particularities, the work's purpose was built as a general framework construction about Internet of Things coherent with the delineated philosophy and that possess an inferential power of knowledge construction.

As a result, was used the Prolog language, as a first-order logic implementation, in order to cause the interconnection and entanglement of the objects used, producing inferences, in other words, relations that were implicit became explicit. In conclusion, to exemplify the framework's power, it was instantiated a basic and minimum application of the framework inside the context of home automation.

Keywords: Internet of Things, philosophy of technology, framework, first-order logic, Prolog.

Sumário

1. INTRODUÇÃO	7
1.3 ESTRUTURA DE APRESENTAÇÃO	11
2. INTERNET DAS COISAS	13
2.1 APRESENTAÇÃO	13
2.2 PROBLEMA E DESVIO	15
3. FILOSOFIA DA TECNOLOGIA	16
3.1 INTRODUÇÃO	16
3.2 ERNST KAPP, MCLUHAN E DAVID ROTHENBERG	16
3.3 PÓS-FENOMENOLOGIA.....	18
4. DE VOLTA À INTERNET DAS COISAS	20
5. FRAMEWORK	25
5.1 DELINEAÇÃO.....	25
5.2 ESP8266 E NODE-MCU	27
5.3 MQTT	30
5.4 MONGO-DB	30
5.5 NODE.JS	31
5.6 PROLOG	31
5.7 APRESENTAÇÃO COM SISTEMA BÁSICO COMPLETO	32
6. ESPECULAÇÕES	37
6.1 LINHAS DE DESENVOLVIMENTO E DE FUGA.....	Erro! Indicador não definido.
6.2 ALAVANCA DIGITAL	37
6.3 GADO HUMANO	39

7. CONCLUSÃO	42
8. REFERÊNCIAS	45
9. APÊNDICES.....	46
9.1 CÓDIGO DO objeto_1.c PROGRAMADO EM C:.....	46
9.2 CÓDIGO DO objeto_2.c PROGRAMADO EM C:.....	51
9.3 CÓDIGO DO BROKER.JS PROGRAMADO EM NODE.JS:.....	56

Lista de ilustrações

Figura 1 - 12 fatores que impulsionam a Internet das Coisas	15
Figura 2 - Relação homem e artefatos, onde os últimos são entendidos como um sistema de objetos correlacionados entre si.	21
Figura 3 - Grafo de extensão do corpo com seus artefatos conectados na forma de extensão de função.	22
Figura 4 - Grafo de extensão mostrando na região hachurada a comunicação via Internet das Coisas.	23
Figura 5 - Equivalência empírico-teórica utilizada onde um módulo ESP8266 NodeMCU corresponde a um objeto qualquer	26
Figura 6 - Esquema do MQTT realçando a funcionalidade de desacoplamento	27
Figura 7 - NodeMCU junto com o ESP8266.....	29
Figura 8 - Esquema representando as propriedades básicas do MQTT	30
Figura 9 - Esquema do sistema básico completo.....	33
Figura 10 - Alavanca física utilizada por Reuleaux para explicar o seu método, junto com a sua solução.	38

1. INTRODUÇÃO

1.1 ANTECÂMARA

O corrente trabalho surgiu de outra origem, de um tópico inter-relacionado, mas de escopo diferente, e de certa forma de um ponto de vista correspondente no qual uma figura do uso da tecnologia da Internet das Coisas se mostrava. “Automação residencial usando Internet das Coisas” era esse tópico, onde algumas diretrizes já se apresentavam como o uso de módulos WiFi, Bluetooth e diversos outros dispositivos microcontrolantes pertinentes. Além de já se ter alguma resolução clara do projeto em si, ou seja, as perguntas não escapariam muito de por exemplo “quantos sensores” ou em quais lugares desse setor residencial eles seriam colocados. A própria implementação em outros projetos com o tema parecido encontrava-se de antemão multe existente. Realizar essa interconexão residencial, embora tenha muitos pormenores importantes, não é um problema quando observada daquele ponto de vista que o próprio tema “Automação residencial usando Internet das Coisas” proporcionava. Era um tópico estritamente em variedade quantitativa entre eles, e não qualitativa, faltava alguma coisa.

Procurar um melhor ponto de observação mostrou-se como um ímpeto necessário – refundar o espaço do projeto um dever.

Revendo o problema de frente tudo parecia um pouco passivo demais, os objetos tanto ficavam esperando que o projetista os introduzisse na rede residencial, quanto era estático: uma vez interligados, e.g. sensor de presença ligado na luz dum cômodo, ficariam nessa configuração até segunda ordem. Da linguagem utilizada também um problema surgia: falava-se de objetos e objetos e objetos, mas raramente qualquer passo de abstração do que eles seriam, ou como seriam era confeccionado. Como exemplo a geladeira falava com a sua conta on-line, mas o software da geladeira-como-objeto era só dela, como uma roupa com único uso, o sistema residencial observava suas partes respeitando demais as ultras diferenças físicas de cada objeto.

Disposições, pequenas heurísticas – farejar o conteúdo da Internet das Coisas sempre apontava numa via cuja direção era de um mínimo de abstração. Não aquela para fora do real, e sim penetrando ainda mais nele. Substancialmente diferentes, esses objetos perdidos e jogados, uma vez dentro dos sistemas, se comportam *realmente* com um ritmo mais simples do que é possível imaginar inicialmente.

Precisa-se de uma mudança de ponto de observação – distância percorrida via inúmeras veredas as quais mostrar-se-ão em breve, na seção sobre filosofia da tecnologia. Mas antes a resolução da “Internet das Coisas” deverá ser aumentada.

Portanto, primeiramente irá se refundar o que realmente se quer e só depois voltar-se-á para Internet das Coisas com uma bagagem e ponto de vista diferentes. Depois uma implementação desse sistema sob essa nova luz será realizada, estimando o quanto das tecnologias disponíveis ajudaram na conjuração empírica da ideia. Produzindo uma potencialidade zero, um nível basal de funcionamento do esqueleto, a saber um framework: a forma do uso, não o conteúdo. Por fim certas especulações em paralelo que se mostraram úteis na externalização e desenvolvimento do projeto serão discutidas, seguida pela conclusão fechando o trabalho.

1.2 MOTIVAÇÕES PRÓXIMAS

Não unicamente a onda da Internet das Coisas é que está trazendo tudo mais próximo e interconectado do que antes. Existem certas tendências em outras disciplinas e campos conceituais não necessariamente dentro da engenharia eletrônica que trazem esse espaço de ideias em foco. Com enfoques diferentes, cada uma dessas entidades conceituais enriquece qualquer toque interdisciplinar – que o presente trabalho tenta seguir – além de deixar em suspenso o quanto a origem delas teve um ancestral comum. É claro que as intersecções conceituais podem parecer mínimas a uma primeira leitura, mas mesmo assim é extremamente produtora tentar observar a surpresa dessas “coincidências” as quais serão serialmente descritas nos parágrafos a seguir, na filosofia da linguagem, sistemas complexos e metafísica naturalista estrutural.

Na filosofia da linguagem foi desenvolvido o subcampo do inferencialismo, doutrina em grande parte fundada pelo filósofo americano Wilfrid Sellars e desenvolvida pelo seu orientado Robert Brandom, ambos embora da escola chamada filosofia analítica, sempre tiveram interesse em figuras do continente como Immanuel Kant e Hegel – ambos alemães. Com uma visão pragmatista do uso da linguagem - ou seja, para investigar as questões fundamentais da linguagem as melhores perguntas são do tipo “o que se faz quando se usa tal conceito” e não simplesmente “qual é o fundamento do conceito tal” – esses filósofos desenvolveram um pragmatismo analítico que se tornou popular nas rodas acadêmicas, tendo até certa projeção em outros autores não familiarizados com a tradição analítica. Para se ter

uma visão do conteúdo sistêmico e de interconexões desse tópico é expressivo citar uma passagem dum livro recente de Brandom:

And it is a rationalist expressivism in that it understands expressing something, making it explicit, as putting it in a form in which it can both serve as and stand in need of reasons: a form in which it can serve as both premise and conclusion in inferences. Saying or thinking that things are thus-and-so is undertaking a distinctive kind of inferentially articulated commitment: putting it forward as a fit premise for further inferences, that is, authorizing its use as such a premise, and undertaking responsibility to entitle oneself to that commitment, to vindicate one's authority, under suitable circumstances, paradigmatically by exhibiting it as the conclusion of an inference from other such commitments to which one is or can become entitled. (BRANDOM, 2009)

Ou seja, essa rede produzida pelas inferências, entre aquelas que entram como premissas e aquelas que saem como conclusão dedutivas formam um campo conceitual de interconexões entre sistemas, porém nesse caso sendo a língua que fornece as partes simples relevantes, que aqui pode ser também observada como um sistema independente com as palavras entrando no lugar dos objetos analogamente.

O campo de sistemas complexos ganha cada vez mais projeção acadêmica e as vezes até fora, como por exemplo em livros de ciência popular. Largamente multidisciplinar ele teve vários nomes e linhas de desenvolvimento que a princípio pareceram paralelas, mas que a posteriori acabaram se aproximando. Dentre outros, tem-se a cibernética, inteligência artificial, teoria do caos, teoria de sistemas, lógica *fuzzy*, redes neurais e modelos computacionais baseados em agente. Portanto não se desenvolveu nenhuma estrutura conceitual coesa e única, tendo diversas definições das quais uma útil ao nosso contexto é:

Complex systems consist of a large number of mutually interacting and interwoven elements, parts or agents defined by the structure of the system, the types of interactions between system elements, and the dynamics and patterns of the system that emerge from these interactions. (MANDUCA, 2006)

Onde esses elementos - na citação - são entendidos como os objetos duma aplicação possível em Internet das Coisas, onde eles também entraram em interação, definindo uma estrutura emergente que não pode ser facilmente reduzida às partes. Pode-se até esperar que certas complexidades dentro dum sistema IoT aumentarão progressivamente a medida que cada subsistema conecta com outro subsistema e assim por diante, tendo resultados imprevistos inicialmente, ou seja, que nenhum engenheiro desses subsistemas correspondentes previu quando no desenvolvimento de cada um deles.

Como último tópico de motivação próxima tem-se a supracitada metafísica

naturalista estrutural, que parece inofensiva se considerarmos apenas como elucubrações filosóficas sem conexão com a realidade empírica. Todavia ela pode embasar uma filosofia da ciência através da qual direções de interpretação podem-se revelar. Basicamente a metafísica tenta especificar a natureza daquilo que existe de mais fundamental, a saber, as entidades básicas do ser. E o predicado naturalista qualifica que essa especificação deve ser centrada na natureza e suas leis empíricas.

Uma das primeiras aparições do que depois iria se chamar realismo estrutural (uma forma da metafísica supracitada) foi delineado no trabalho do matemático e filósofo francês Henri Poincaré, onde realçava a importância das relações entre as entidades em vez de perguntas sobre o que cada entidade era no seu âmago, ou seja, a natureza dá dicas somente sobre sua estrutura e interrelação entre suas partes. Qualquer reflexão das coisas-em-si é contraproducente e sempre acaba produzindo erros – vide a mudança nas teorias científicas que conservam certa estrutura, mas mudam totalmente o entendimento da matéria-em-si.

Uma abordagem mais rigorosa e contemporânea foi dada pelos filósofos James Ladyman e Don Ross:

Ontic Structural Realism (OSR) is the view that the world has an objective modal structure that is ontologically fundamental, in the sense of not supervening on the intrinsic properties of a set of individuals. According to OSR, even the identity and individuality of objects depends on the relational structure of the world. Hence, a first approximation to our metaphysics is: 'There are no things. Structure is all there is. (LADYMAN, ROSS, 2007)

Se a estrutura é tudo o que existe, o presente trabalho está na direção correta, i.e., focar-se-á na rede dos objetos, o que eles fazem quando se interligam, seus tópicos de conversa, e não se prestará atenção nas vicissitudes e particulares empíricas de cada um deles. O passo é um de abstração.

1.3 ESTRUTURA DE APRESENTAÇÃO

A estrutura de apresentação do argumento narra-se através primeiramente da exploração conceitual da Internet das Coisas (seção 2), ou seja, o que se pode referir na linguagem de sistemas a essa categoria. Após isso, a fertilização conceitual pela filosofia da tecnologia será realizada na seção 3, com a consequente volta para o tópico inicial com uma nova bagagem na seção 4. Depois dessa requerida análise dos conteúdos, na seção 5, nomeada framework, a delineação do sistema e sua instanciação com respectivas tecnologias relevantes será apresentada num caso de estudo sobre automação residencial. Por fim, na seção de especulações, algumas

tendências e ideias ainda não exploradas, mas pensadas, e que são pertinentes para o projeto, serão apresentadas. Fecha-se na seção final, a conclusão, com o resultado, e uma correspondente meta-conclusão, ou seja, o que se aprendeu não diretamente sobre o projeto, mas sobre as tendências de abordagem de um tópico futuro qualquer que não necessariamente teria ligação conceitual com a Internet das Coisas.

2. INTERNET DAS COISAS

2.1 APRESENTAÇÃO

O incrível pensamento da inicial grande popularidade da assim chamada Internet das Coisas é o seguinte a priori cristalino argumento, colocado em formato de um raciocínio lógico com 2 premissas e uma conclusão:

- 1) Bilhões de “coisas” estão aí perigando sem conexão com a internet, sejam itens ubíquos desde de máquinas de suco até o carro particular. Desejariam muito dos seus senhores a permissão de entrada na rede.
- 2) Tecnologias de conexão embarcadas e.g. módulos WiFi microcontrolantes ou RFID (*radio frequency identification*) estão cada vez mais baratas e ultrapassando o peso do custo da implantação, chegando até mesmo parecer sair “de graça” quando se leva em conta todo o seu potencial de agregação de valor ao produto.
- 3) Portanto esses bilhões de entidades entrarão por alguma razão na articulação das linguagens residenciais, industriais, cívicas, dentre outras, apenas dependendo da vontade de interligá-las.

Sim, bem parece um nova estratégia sub-reptícia de venda de consultorias ou outra degeneração de marketing (até porque talvez seja, pelo menos na curva inicial de espalhamento da ideia e construção da imagem). Toda a narrativa do El Dourado das Coisas esperando para ser minado realmente se baseou em certas tendências das aplicações tecnológicas industriais e microssistemas como de casas individuais em países ricos. Tem um fundo fidedigno no final das contas o qual não é esgotado depois que se retira o puro interesse midiático.

É de se notar o fato de que o leitor pode escapar e tentar buscar alguma outra definição um pouco menos duvidosa, ou milimetricamente mais precisa, mas não encontrará. De artigos das implementações com alguns modelos de referência de arquitetura úteis ou livros menos técnicos descrevendo casos de uso, o que se apresenta é uma conveniente expressão: “objetos sendo conectados”. E quando esses objetos são na verdade indivíduos empíricos, o ultra controle não crítico simplesmente lançado como IoT é deveras desconfortável o qual será melhor desenvolvido na seção 6. Ressaltando ainda inúmeros pequenos projetos que são

sempre descritos num formato preditivo de como o dia a dia dum indivíduo daqui a 10 anos, em plena ubiquidade de sistemas Internet das Coisas, se torna ultra conectado e diferente, inundando a vivência comum com novas práticas, e.g. não se preocupar mais se falta tal item de comida no estoque residencial já que o próprio sistema residencial cuidará de orquestrar sua reposição.

Mesmo com essas ressalvas, tantas aquelas tendências supracitadas e outras implementações que só posteriormente foram etiquetadas como Internet da Coisas realmente dão uma tendência bem clara de mudança geral no formato de como praticamente tudo é ligado em redes. Como exemplo de sistema que a princípio não era Internet das Coisas, mas posteriormente foi-se interpretado com um, são os sistemas de consciência de situação:

Today businesses and government are already using enhanced situational awareness to make to manage traffic. Law enforcement agencies in Los Angeles, New York, Memphis, and Santa Cruz are using data input and real-time analytics to pinpoint probable risk and assign officers and resources based on the emerging concept of predictive policing. (GREENGARD, 2015)

Ou o paradigma de ecossistemas Internet de Tudo, o qual é dirigido de maneira distribuída entre os diversos usuários de um sistema. Por exemplo quando utiliza o aplicativo no celular de seus diversos usuários para alimentar um banco de dados na nuvem. Possuindo assim diversos “ouvidos” através dos sensores de cada aparelho, tendo obviamente a aceitação prévia dos clientes para essa exploração. Essa ativa colaboração é manufaturada com apelos de gamificação (montagem de objetivos com incentivos de pontos e títulos) e outros métodos de oportunismo bilateral entre cliente e sistema. (ZIOUVELOU, et al., 2017)

Outra interessante e produtiva aplicação é o monitoramento de paciente com problemas crônicos em tempo real. Em vez de consultas durando minutos em alguns dias na semana, o fluxo de interpretação baseado no estado do paciente é acompanhado em tempo real, sem necessidade de contatos físicos diretos os quais ocasionariam perda de tempo e desgaste do próprio paciente (GREENGARD, 2015)

Poder-se-ia enumerar diversas aplicações com semelhanças de família e o quanto estão se tornando comuns nas implementações tecnológicas, porém as diferenças são quantitativas, e a função de delineamento do que se entende como Internet das Coisas realizada, já é suficiente para os propósitos correntes. Concluindo a esquematização sinteticamente tem-se a seguinte figura mostrando os 12 fatores

com os quais a IoT está se potencializando e ganhando o mundo:

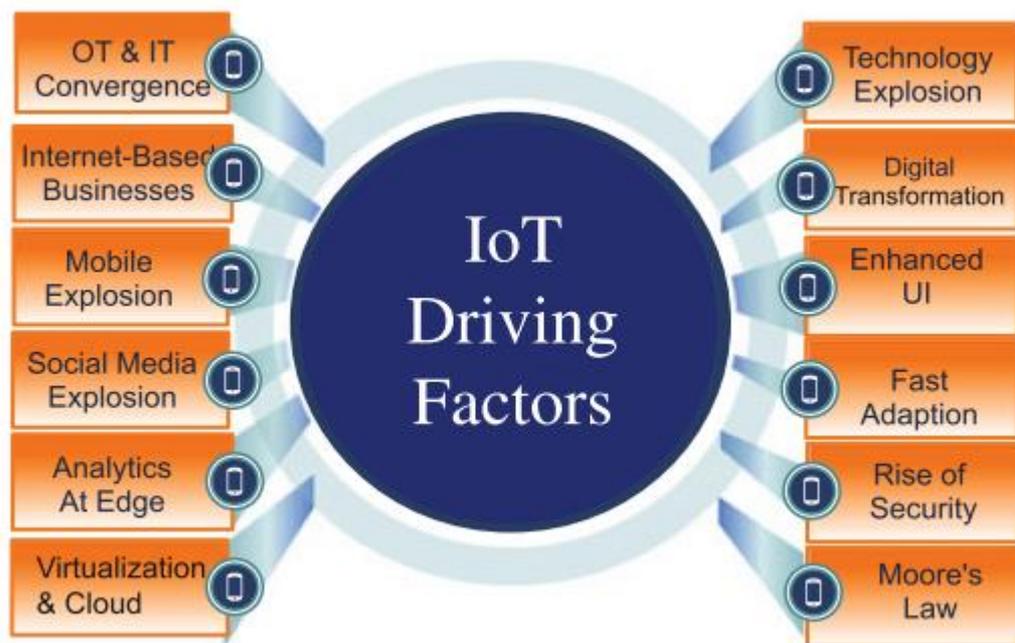


Figura 1 - 12 fatores que impulsionam a Internet das Coisas
Fonte: Rayes e Samer (2017)

2.2 PROBLEMA E DESVIO

Todas essas disposições investigadas na seção 2.1 apontam para uma necessidade de reavaliação da formulação inicial sobre Internet das Coisas. Beiram uma espécie de problema na própria definição, especificamente uma falta de qualquer rigor ou precisão na delineação do que realmente está se falando quando se conversa sobre Internet das Coisas. Claramente poder-se-ia retrucar que é pedir demais de um conceito já abrangente, ou seja, essa falta de mínimo rigor é propriedade da própria IoT. E é exatamente isso que foi duvidado e reavaliado no presente trabalho. Sendo a linha de desvio pela filosofia da tecnologia que será discutida na seção 3.

A necessidade desse desvio surgiu da pura necessidade. Descorrelacionado totalmente duma pura releitura ou análise em paralelo com a implementação. Ao contrário, a presente excursão mudará exatamente os parâmetros da implementação. Sem essa fuga o próprio começo do projeto seria radicalmente diferente, produzir-se-ia somente mais um subsistema em diferença quantitativa com outros da literatura e engenharia eletrônica das aplicações em grande escala.

3. FILOSOFIA DA TECNOLOGIA

3.1 INTRODUÇÃO

Na reflexão dos fundamentos do que é tecnologia, a fertilização do conceito se fez, e suas correspondentes inferências que não eram vistas anteriormente acabaram ajudando no problema o qual teve sua coloração transmutada.

Na introdução e na seção 2.2 sobre os problemas na definição da Internet das Coisas foi mencionada um desnorтеio sobre aonde esse tópico estava apontando. Todos os pequenos casos de implementação, seja uma residência inteligente ou um paciente cardíaco monitorado remotamente pelo seu médico, mostravam apenas uma semelhança de família, mas nenhum princípio debaixo da superfície com o qual se pudesse generalizar o problema.

E vista disso, explorar-se-á quais tipos de respostas são obtidas quando se pergunta os fundamentos do que é um artefato da técnica.

3.2 ERNST KAPP, MCLUHAN E DAVID ROTHENBERG

Os três filósofos Ernst Kapp, Marshall McLuhan e David Rothenberg foram utilizados primeiramente no esmiuçamento livre dum possível fundamento dos artefatos técnicos, todos os três têm respostas diferentes, porém com crescimento progressivo de sofisticação – vide a ordem cronológica do surgimento respectivo dos trabalhos de cada um (BREY, 2000). Explorar-se-á os três a seguir.

Ernst Kapp (15 de outubro 1808 – 30 de janeiro 1896) foi um filósofo e geógrafo alemão, cujo trabalho foi pioneiro na filosofia da tecnologia, praticamente fundando o campo de estudo, com ideias extremamente semelhantes às de Marshall McLuhan, porém com quase cem anos de vantagem. Nos seus textos de filosofia ele apresentou inúmeros argumentos a favor do entendimento dos artefatos técnicos como projeções dos órgãos humanos. Apresentando a tese de que os humanos inconscientemente transferem a forma, a função e as proporções normais de seu corpo para os artefatos de suas mãos (KAPP, 1877).

Kapp apresenta notável variação de pareamentos de órgão-artefato entre os quais similaridades morfológicas – como mencionado anteriormente – existem, aliás com origem de forma consciente. O dedo encurvado torna-se um anzol, o oco da mão

uma tigela, o braço e a mão uma pá, os nervos cabos telegráficos, sistemas de trem o sistema vascular, e assim por diante.

Muito imaginativo, todavia apresenta algumas limitações, até possivelmente históricas dado que tinha em mãos as tecnologias do século XIX. Sua teoria não explica muito bem, ou simplesmente deixa de considerar outras tecnologias que não apresentam emparelhamento morfológico claro com livros, telefones, aviões e outros diversos itens.

Já Marshall McLuhan (21 de julho, 1911 – 31 de dezembro, 1980) – filósofo canadense - sofisticou a linguagem de Kapp retirando a necessidade dos artefatos terem necessariamente semelhanças ou origem na morfologia humana. Ou seja, somente propriedades funcionais dos órgãos humanos que são traduzidas, em forma amplificada, para os artefatos. Além disso, McLuhan introduz uma gramática das extensões, em outras palavras, os artefatos estenderiam o poder e o escopo de cada potencialidade humana, sejam elas o sentido, o sistema nervoso central ou até a consciência. Como McLuhan exemplifica de sua maneira única:

‘The city, like a ship, is a collective extension of the castle of our skins, even as clothing is an extension of our individual skins. But weapons proper are extensions of hands, nails, and teeth, and come into existence as tools needed for accelerating the processing of matter’. (MCLUHAN, 1964)

Depois de cerca de 30 anos após a publicação dos trabalhos mais importantes de McLuhan, outro filósofo, o americano David Rothenberg, aprofundou suas ideias argumentando que a tecnologia pode estender todos aqueles aspectos humanos os quais há possessão do entendimento mecanicista, que se sabe como funciona. Já outras práticas como as faculdades da moralidade, do julgamento e o senso de pertencimento, onde esse conhecimento não existe, não podem ser estendidas.

Dentro daquelas faculdades que podem ser estendidas, Rothenberg as categorizou em duas classes: faculdades de ação e faculdades de pensamento. A primeira, as de ação, correspondem as extensões diretas mais simples e corporais, de ações manuais, como as de transporte (veículos). Já as de pensamento são as extensões dos sentidos, sistema nervoso central e consciência, como linguagens naturais e formais, programação computacional, calculadoras, dentre outras.

Continuando o fio do presente trabalho, como poder-se-ia conectar essa nova visão dos artefatos como extensão com o presente grupo de ideais da Internet das Coisas? É exatamente isso que ocasionou o ponto de virada do projeto, para outra vereda, e é o que explicar-se-á na seção 4.

3.3 PÓS-FENOMENOLOGIA

A análise filosófica da seção 2 já possui substância suficiente para ocasionar a mudança de perspectiva que estava se procurando desde o início. Todavia, explorando mais esse tópico foi importante o estudo de uma tradição do pensamento com desenvolvimento recente que proporcionou uma notação e uma linguagem um pouco diferente, mas vital para descrever melhor o que se deseja. É a chamada pós-fenomenologia, inaugurada pelo filósofo americano Don Ihde através da publicação do livro *Technology and the Life World* (1990), no qual realizou uma releitura dos autores principais da tradição da fenomenologia como Martin Heidegger e Maurice Merleau-Ponty, adicionando conceitos pragmáticos e de análise empírica dos subcasos de aplicação tecnológica.

Uma delimitação extremamente útil que Don Ihde produziu foi a divisão das diversas tecnologias em padrões estruturais de uso nos quais nossa interação incorporada muda de acordo com o tipo. As três divisões assim diferenciadas são as técnicas de incorporação, as técnicas de hermenêutica e as técnicas de relações de alteridade, explicadas abaixo.

Técnicas de incorporação são aquelas tecnologias que passam a fazer parte do próprio corpo do usuário assim que seu uso é acostumado, realizando seu retraimento da consciência ativa do usuário de tal forma que se passe a perceber o ambiente através dessas tecnologias, ou seja, ela atua numa posição de mediação, um claro exemplo são os óculos. Formalizando essa relação, e é aqui que a principal razão da excursão nesse tópico teve sua razão de ser, tem-se:

$$(Eu\text{-}tecnologia) \rightarrow \text{mundo} \quad (3.1)$$

Técnicas hermenêuticas são aquelas utilizadas para transformar o ambiente num esquema no qual ele melhor possa ser lido e assim tendo seu sentido explicitado. A diferença das de incorporação é que agora o foco da atenção não está num objeto no ambiente do qual a tecnologia amplificará suas características, como num telescópio. Mas sim no texto ou no painel, como na linguagem escrita, e assim a conexão se torna semi-opaca e produtora de sentido. A interpretação é originada da conexão semântica do referente ao referenciado. Formalizando essa relação:

$$\text{Eu} \rightarrow (\text{tecnologia-mundo}) \quad (3.2)$$

Técnicas de relações de alteridade é quando a tecnologia sendo utilizada ganha um aspecto de quase outro ser, quando se torna complexa demais – aqui é bom imaginar uma inteligência artificial – e a relação com esse artefato torna-se uma pequena criação de um espaço de subjetividades a parte. Não é simplesmente uma maneira de ler um fenômeno como a hermenêutica anterior, esse tipo sofisticado de prática cria aquilo o que está sendo visto de uma maneira radical. Formalizando esse subtipo tem-se:

$$\text{Eu} \rightarrow \text{tecnologia-}(-\text{mundo}) \quad (3.3)$$

Com essas notações formalizadas ficará mais simples delinear onde estaria tal objeto em relação ao seu usuário, como ficará evidente na seção 4 essas interconexões expõem as propriedades do sistema de maneira natural e direta de tal forma que o passo de abstração, uma vez externada numa notação parecida, se torna praticamente um passo lógico sobre a reflexão do que foi montado até aquele momento.

4. DE VOLTA À INTERNET DAS COISAS

De posse desse novo entendimento do que é tecnologia, ou melhor, tendo-se melhor embasamento dos possíveis aspectos das relações entre o homem e um artefato, pode-se especular, de maneira um pouco ingênua, que aparentemente nada dos esquemas de pensamento sobre a Internet das Coisas realmente mudou. Exatamente esta posição que o argumento nega enfaticamente. Para isso, imagine a relação básica dum sujeito com um artefato, primeiramente uma relação de incorporação delineada na equação 3.1. Desconsiderando maiores aspectos dessa tecnologia, se é um marca-passo ou óculos, chame-o de objeto, um O_1 em notação básica. Portanto tem-se a seguinte relação inicial:

$$(Eu-O_1) \rightarrow \text{Mundo} \quad (3.4)$$

Supõe-se até a consideração de um outro objeto O_2 , numa relação hermenêutica, sendo que ignorar-se-á as de alteridade por serem tecnologia não tão bem desenvolvidas:

$$Eu \rightarrow (O_2\text{-Mundo}) \quad (3.5)$$

A posição da seta \rightarrow , em contrapartida aos parênteses, é só um auxílio para o pensamento, as conexões do Eu aos objetos continuam existindo num formato primário, ou seja, pode-se esquecer a seta e utilizar apenas o traço para facilitar o modelo de entendimento.

Contudo, como se deve ter percebido pela numeração proposital na notação dos objetos, esse é um raciocínio indutivo. O adcionamento de mais artefatos, a princípio, porém com desincentivos práticos, pode ser continuado em proporção com uma determinada aplicação ou uso do modelo específico de tal problema. Para fins expositivos colocar-se-á mais um objeto O_3 – podendo ser do tipo de incorporação embora não importe mais.

Como tem-se apenas um Eu, um homem, os 3 objetos se conectariam diretamente no sujeito o qual seria uma espécie de praça central na cidade homem/artefatos. Nesse ponto o truque da IoT entra em cena: esses 3 objetos também se comunicarão entre si, cada morador numa casa nessa cidade não necessita se locomover até a praça, lá achar a locação do seu destino, e só por

consequente atingir seu objetivo – obviamente essa metáfora conjura os pequenos cidadãos como pedaços de informação no fluxo incessante nesse microssistema. Há um desafogamento da carga cognitiva sobre o Eu, qualquer tarefa de triangulação pode ser feita entre os dois objetos em vez de ter de chamar a todo momento, as vezes de maneira intrusa, a atenção/foco do usuário – a geladeira manda uma lista automaticamente no celular dos produtos que necessitam de compra, o Eu não precisa inspecionar a geladeira e passar para essa sua lista digital. Sintetizando num diagrama, mostrado na Figura 2, pode-se delinear o seguinte:

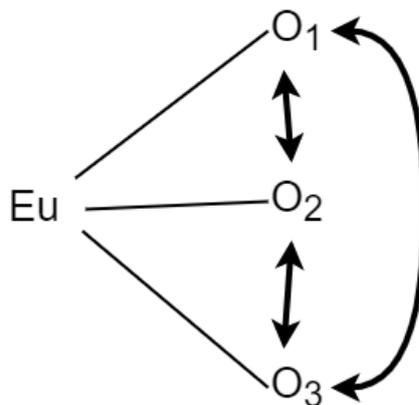


Figura 2 - Relação homem e artefatos, onde os últimos são entendidos como um sistema de objetos correlacionados entre si.

Mais do que isso, perceba que uma conexão singular com um objeto engendra uma correspondente capacidade perceptiva ou cognitiva estendida ou empoderada. Reescrevendo de outra forma, a própria unidade do Eu exemplificada nas relações supracitadas pode ser destrinchada e melhor elaborada – explicitando essas relações específicas. Precisamente esse método que o diagrama mostrado na Figura 3, chamado de grafo de extensão apresenta e exemplifica, utilizando uma notação aonde o Eu é analisado em seus subsistemas somente a título de instanciação, sem o intuito de ser exaustivo ou biologicamente preciso. Junto com a indicação de qual artefato estaria estendendo qual subsistema do Eu:

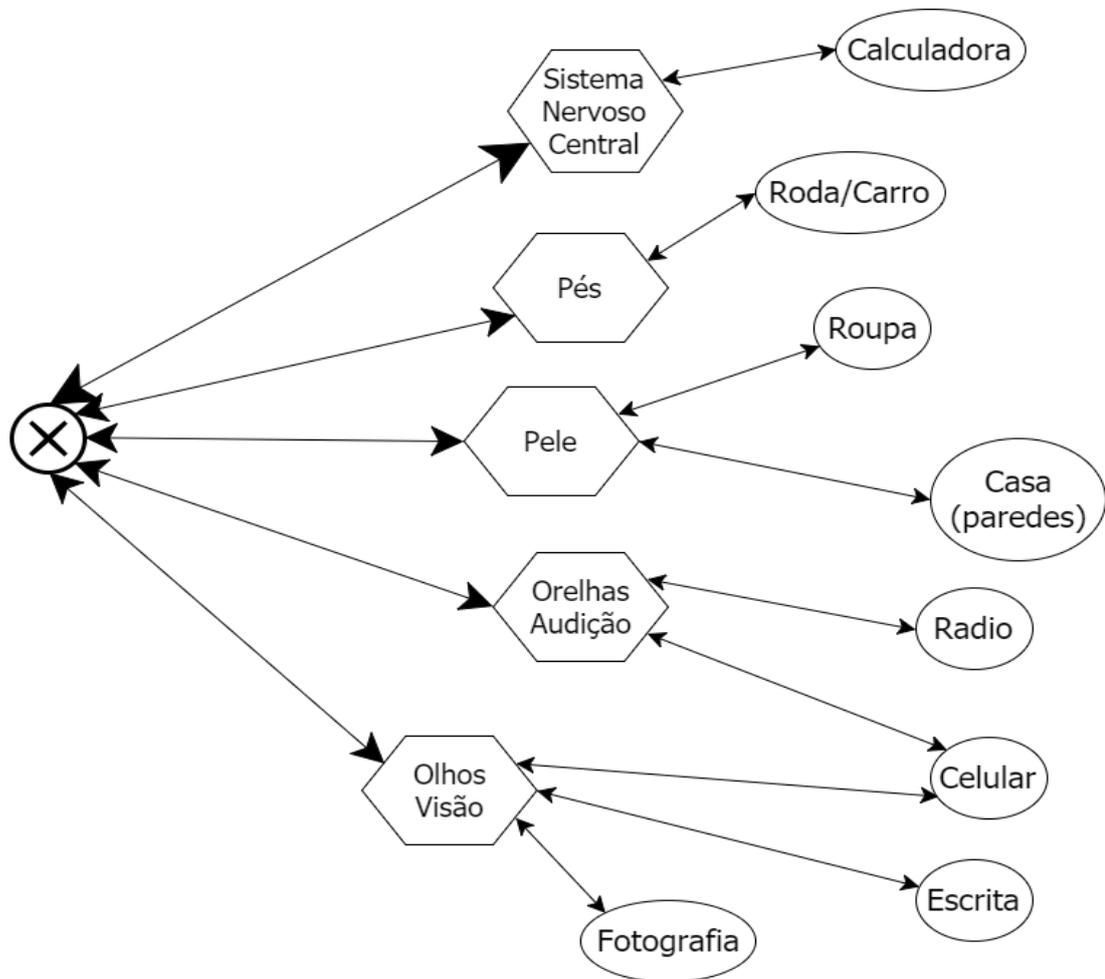


Figura 3 - Grafo de extensão do corpo com seus artefatos conectados na forma de extensão de função.

Realizando agora o mesmo passo teórico anterior para os objetos dos quais outras conexões foram realizadas, pode-se conjurar – e aqui é o cerne do raciocínio inteiro – um espaço de intercomunicação entre essas entidades, o que seria, em outras palavras, a eficácia e o princípio da IoT:

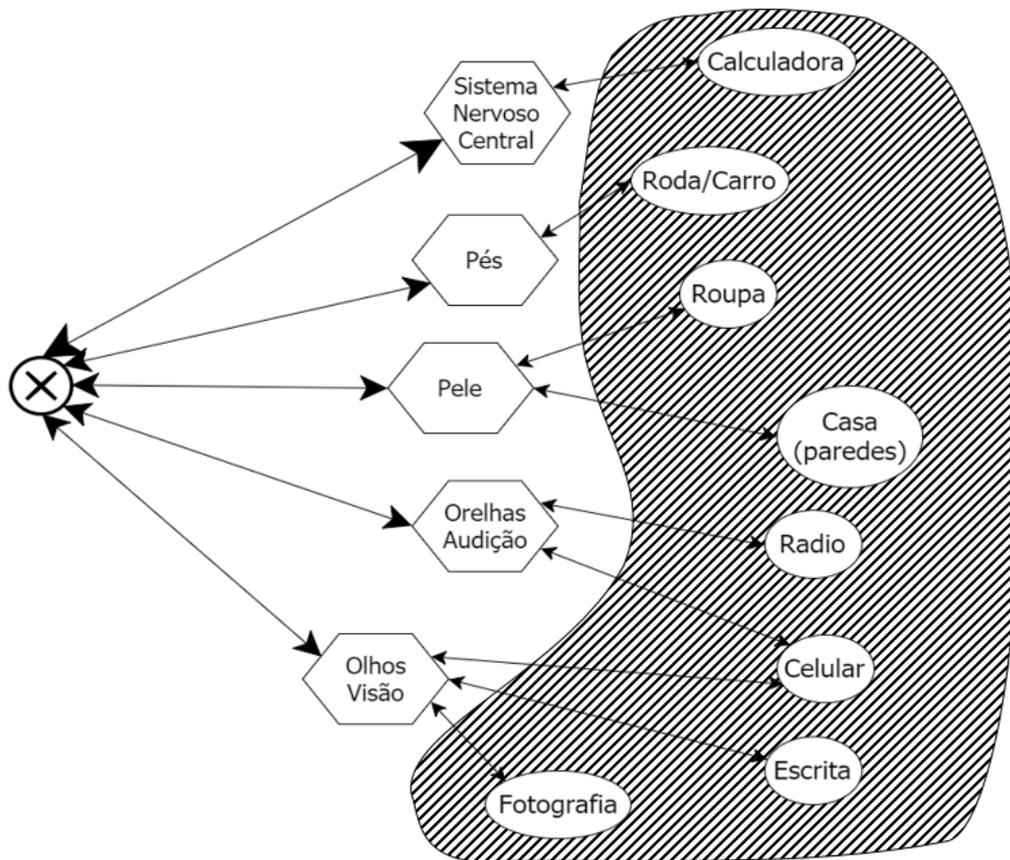


Figura 4 - Grafo de extensão mostrando na região hachurada a comunicação via Internet das Coisas.

Observe que a interrelação dos objetos funciona de maneira análoga com a interrelação entre os próprios subsistemas tradicionais do corpo. Ou seja, o sistema nervoso e o sistema digestivos conversam entre si, a saber via conexões nervosas ou hormonais (a precisa descrição dessa comunicação não é o escopo aqui, só a sua manifesta existência) numa maneira que pode ser aproximada – obviamente numa escala muito menor de complexidade – pelos objetos no formato de artefatos tecnológicos. E qual seria o uso desse jogo de analogias? A resposta é que a mesma funcionalidade das linguagens de intercomunicações fisiológicas do organismo as quais trabalham para o benefício do Eu sem que necessite de sua atenção e direcionamento, será a do sistema dos objetos. Eles também realizarão tarefas automáticas tornando-se uma entidade corporal única.

E é essa interpretação que a IoT gera no seu uso que é qualitativamente diferente de apenas um grupo ou família de semelhanças prototípicas entre implementações de sistemas de objetos. Sendo exatamente esse ponto que é esquecido nas análises sobre as implementações IoT, sem perceber, que existe sim um alfinete conceitual o qual segura essas definições escorregadias no lugar.

Terminando essa renovação da abordagem, o próprio problema se mostra como necessitando de um dinamismo dentre os objetos, ou seja, que no momento que o sistema mude com a adição de um novo objeto, que as mudanças sejam feitas com certa lógica – adaptando e fornecendo o que pode ser útil ou não ao novo cidadão da cidade. Quer-se a seguinte operação Π , mostrada na Equação 3.6, onde S_n é um sistema qualquer de n objetos e o O_{n+1} o mais novo objeto – a qual será implementada na seção do Framework.

$$\Pi(S_n, O_{n+1}) = S_{n+1} \quad (3.6)$$

5. FRAMEWORK

5.1 ESTRUTURA DE APRESENTAÇÃO

Será primeiramente explicado como o framework pode ser utilizado em uma aplicação direta e simples dentro de um contexto de automação residencial. Contendo um objeto representando uma fonte de luz junto com um interruptor: objeto 2, e um objeto representando um sensor de presença dentro do mesmo quarto: objeto 1.

Para esse fim será apresentado o desenvolvimento em cada tecnologia necessária para essa aplicação, como também uma delineação explicando a razão do uso de cada uma.

Todavia, note que essa aplicação do framework não é o resultado final do framework em si, pois ele possui 3 níveis:

- 1) Framework geral de Internet das Coisas sem nenhuma tecnologia específica selecionada. Possuindo o mínimo possível de diretivas, e flexível o bastante para englobar as diversas aplicações. Esse é o alvo principal e objetivo do trabalho inteiro.
- 2) Uma instanciação do framework em tecnologias adequadas, podendo possuir uma potencialidade de escalamento (como o uso de um banco de dados). Por exemplo um sistema de automação residencial com potencialidade de adicionar novos objetos através da equação (3.6).
- 3) Uma aplicação específica dentro da instanciação do framework em 2) de tal forma que os objetos e seus tópicos de conversa já existam. Nesse caso, o framework entraria na interligação lógica e comunicação entre os objetos desse exemplo. Pode-se citar o quarto com o interruptor, lâmpada e sensor de presença, delineado nas seções 5.2, 5.3, 5.4, 5.5, 5.6, 5.7 e 5.8.

5.2 DELINEAÇÃO DA INSTANCIAÇÃO DE UMA APLICAÇÃO

Precisa-se de uma implementação que seja simples, de tal forma que se possa instanciar as ideias descritas nas seções anteriores, mas que não introduza complicações sobre o aspecto particular de um equipamento físico. Isto é, o objeto

como microcontrolador utilizado deve ter as características de comportamento mínimas de uma entidade embarcada. Sendo um modelo de múltipla função de maneira que se futuramente se queira projetar um sistema, esses mesmos módulos simples de processamento sejam manipulados e realocados na nova função sem maiores mudanças paradigmáticas.

A força da abstração que necessita desses encapsulamentos é necessária quando se precisa criar um sistema basal com o qual outros mais específicos serão construídos em cima. O intuito é prover o esqueleto - como comentado na seção 1.2 o enfoque na estrutura é benéfico pois divide o tempo de trabalho entre os projetistas. Aquele que começará a desenvolver uma implementação tal, não precisará pensar tudo do início, já terá um esquema no qual o seu pensamento pode se direcionar.

Tendo essas ressalvas em mente ter-se-á um módulo ESP8266 NodeMCU (que será melhor explicado em breve) representando um objeto qualquer, ou seja, a relação é pintada da seguinte forma, mostrada na Figura 5:

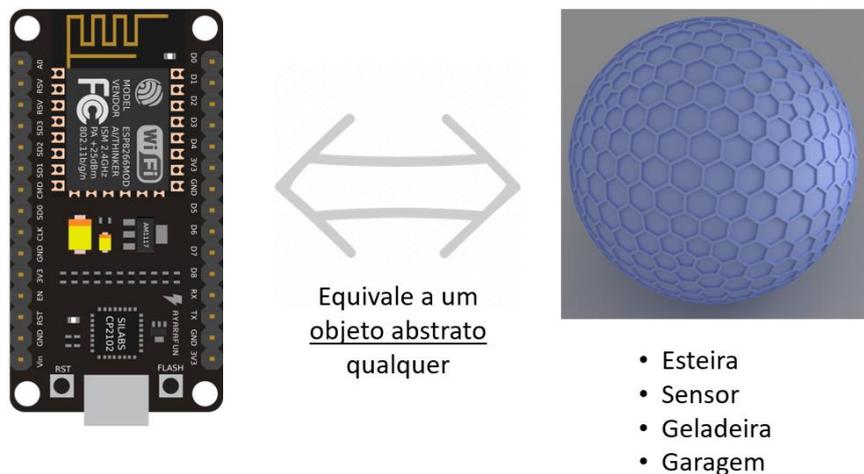


Figura 5 - Equivalência empírico-teórica utilizada onde um módulo ESP8266 NodeMCU corresponde a um objeto qualquer

Além dessa entidade, sua intercomunicação utilizará o protocolo MQTT (HUNKELER et al., 2008), ideal para implementações que necessitam de desacoplamento (como mostra o esquema abaixo) em comunicação de máquina para máquina, além de possuir o paradigma de *publisher/subscriber* – facilitando a implementação em Internet das Coisas– maiores delineamentos na seção 5.3.

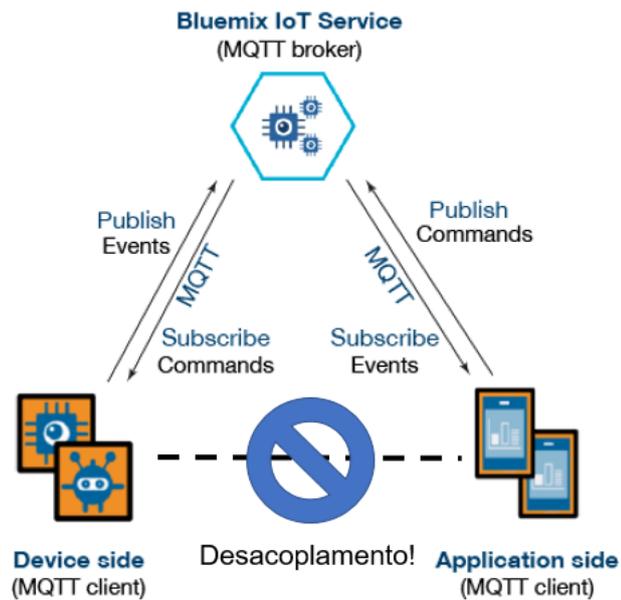


Figura 6 - Esquema do MQTT realçando a funcionalidade de desacoplamento
 Fonte: <https://www.slideshare.net/EyesOfThings>, editado

Com isso, pelo uso do MQTT, claramente se necessitaria de um *broker*, o qual será implementado num server próprio. No caso dessa implementação específica no laptop do autor, mas que, com o código portátil, poder-se-ia ser relocado e estocado em qualquer outro lugar com acesso à Internet. O controle do *broker* é essencial visto que as informações dos entrelaçamentos estão implícitas dentro dele, com o log das entradas e saídas, publicações e inscrições nos tópicos.

O código usado na orquestração é via a linguagem JavaScript (FLANAGAN, 2006) através do ambiente Node.js, adaptado ao processamento a eventos, uma necessidade em unidades embarcadas escutando via sensores, por exemplo. O seu uso será no controle da parte do broker, com um banco de dados MongoDB (CHODOROW, 2013) para esquematizar um escalamento de recursos em grandes possíveis sistemas.

Por fim a extração da informação será realizada na linguagem lógica de programação Prolog depois das relações serem extraídas do Broker e exportados já escritas na sintaxe relevante.

5.3 ESP8266 E NODE-MCU

Várias vias apontam para um módulo de WiFi, um mínimo cordão umbilical conectado com a rede Web é necessário para começar a falar sobre interconexões e sistema de objetos. Sendo inúmeras as opções, certas heurísticas foram usadas, uma delas é o óbvio direcionamento via o uso: quais módulos de desenvolvimento já existem cujo desenvolvimento possui uma comunidade embebida nas aplicações das Internet das Coisas. Resposta: o chip de baixo custo com funcionalidade TCP/IP e unidade micro controladora chamado ESP8266, produzido pela fabricante chinesa *Expressif Systems*, o qual progressivamente tornou-se conhecido pelos desenvolvedores do Oeste devido ao seu incrível baixo custo e pequeno número de componentes externos – apontando para produção em volume.

Configurações do ESP8266:

- Microprocessador L106 32-bit RISC baseado no “Tensilica Xtensa Diamond Standard 106Micro” possuindo clock de 80MHz.
- 64 KiB de RAM de instrução, 96 KiB de RAM de dados
- Flash QSPI externo: 512 KiB até 4 MiB (suportando até 16 MiB)
- IEEE 802.11 b/g/n Wi-Fi
 - Botão TR integrado, balun, LNA, amplificador de potência e rede de casamento
 - Autenticação WEP ou WPA/WPA2, ou rede livre
- 16 GPIO
- SPI
- I²C
- I²S interfaces com DMA (compartilhando pinos com GPIO)
- UART em pinos dedicados, incluindo um UART que pode ser ativado em GPIO2
- 10-bit ADC de aproximações simultâneas

Da própria necessidade de uso do ESP8266 e logo após o seu lançamento, as bases do que viria a se tornar o NodeMCU delineou-se. Os dois passos iniciais foram o compartilhamento na rede de desenvolvedores GitHub do código de firmware do NodeMCU em 13 de outubro de 2014, e dois meses depois o projeto foi expandido aglomerando uma plataforma de hardware livre quando o desenvolvedor Huang R compartilhou o formato gerber (de descrição de circuitos impressos) duma placa ESP8266 chamada devkit v0.9. Mais tarde, naquele mesmo mês, outro programador

introduziu a biblioteca do protocolo MQTT que será comentado na seção 5.3, com ainda outro proporcionando o uso de drives LCD, Screen, OLED e telas VGA. Todavia o projeto foi abandonado pelos seus criadores e por conseguinte continuado por um grupo independente de contribuintes os quais cuidaram bem do ritmo do projeto tendo até 40 módulos diferentes até o verão de 2016. A Figura 7 apresenta o NodeMCU junto com o ESP8266.

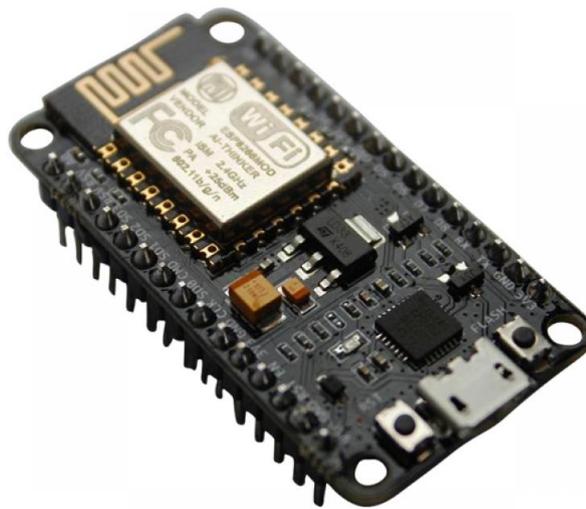


Figura 7 - NodeMCU junto com o ESP8266.

O NodeMCU é uma plataforma para IoT de código livre, ou melhor um SDK (software development kit). Possui um firmware que roda no ESP8266 WiFi SoC (system on chip), e também usa a linguagem de script Lua. Todavia, por razões de facilidade de desenvolvimento e reuso de código não foi explorada a programação via Lua, e sim os programas nos respectivos módulos ESP8266 utilizaram a linguagem C, onde o NodeMCU entra na transação e implementação do código. Criando um ótimo ambiente de testes imediatos de protótipos de acordo com a necessidade de mudança do código.

Além dessa mudança para a linguagem C, a Arduino IDE (Integrated development environment) acompanhada com uma função de “load” do código via cabo USB - como se é acostumado em placas de desenvolvimento - foi utilizada. Causa-se com isso um ambiente produtivo tanto de teste como de própria implementação, onde a edição do código apresenta-se numa distância curta do teste de funcionamento e posterior construção de correlação entre os diversos processos

os quais ficarão mais numerosos de acordo com o adição de mais funcionalidades – como a seção 5.5 sobre o Node.js abordará.

5.4 MQTT

MQTT é um protocolo de conectividade máquina para máquina implementado em cima dos protocolos TCP/IP. Ele é ideal para conexões com locais remotos onde há restrições de tamanho do código e largura de banda da comunicação, como aplicações de dispositivos móveis, ou até em sensores se comunicando via link de satélite sob conexões ocasionais com provedores de sistemas de saúde. Ele foi feito utilizando o paradigma de publicação e subscrição, onde um tópico, alocado no *broker* (o centro onde a orquestração do fluxo de informações é feita), pode ser subscrito, ou seja, toda vez que algo for publicado nesse tópico, todos os clientes subscritos nele receberão o pacote publicado. Um cliente pode publicar em determinado tópico, por exemplo, se algo chegou nos seus sensores, assim, todos os clientes subscritos nesse tópico, receberão esse novo pacote, fechando o ciclo de comunicação. A Figura 8 exemplifica essa dinâmica de comunicação:

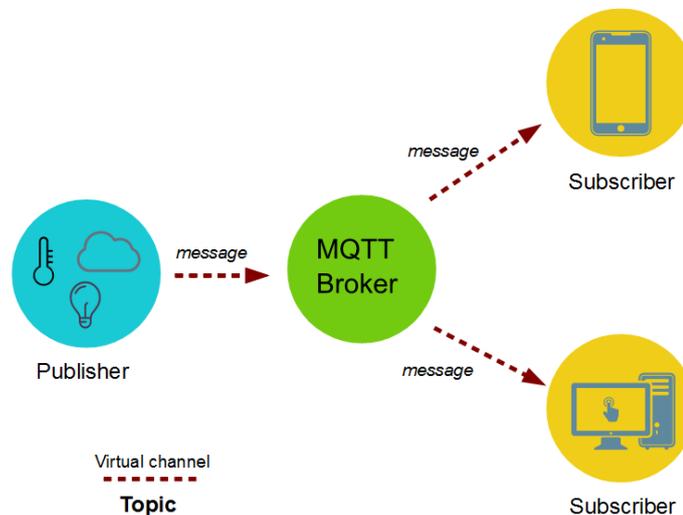


Figura 8 - Esquema representando as propriedades básicas do MQTT
Fonte: <https://www.javacodegeeks.com>

5.5 MONGO-DB

MongoDB é um programa de banco de dados de código livre, orientada a documentos, multiplataforma que utiliza formatos JSON (JavaScript Object Notation), padrão de dados para objetos. Foi utilizado por ser altamente escalável e ter diversas facilidades em formato de bibliotecas para o Node.js nas quais as transações do Broker podem ser gravadas diretamente nesta ferramenta de banco de dados, deixando todo o processo mais coerente e código mais limpo. Criando uma possibilidade de exportação e gerenciamentos dos dados mais eficazes.

5.6 NODE.JS

Um sistema embarcado, principalmente aqueles com pesada carga de processamento digital, não pode permanecer subordinado às particularidades de um processador com um processamento serial demais, ou simplesmente que o programa fique parado numa linha de código enviando um pedido para interrupção em algum periférico. Além disso, o framework do sistema do presente trabalho tem necessidade de algum paradigma aproximado de programação paralela, ou melhor, possua sensibilidade a pedidos com eventos.

A pesquisa de qual linguagem de programação e correspondente ambiente a ser utilizado não foi muito demorada, visto que diversas aplicações para Internet das Coisas a utilizam e inúmeras bibliotecas estão disponíveis. Esse ambiente escolhido é o Node.js, uma plataforma código-livre JavaScript em tempo de execução para executar códigos no lado do servidor.

5.7 PROLOG

Prolog é uma linguagem de programação lógica de fim geral muito utilizada em aplicações de inteligência artificial e computação linguística – dentre elas pode-se citar prova de teoremas automática e sistemas especialistas. Uma de suas marcas é o fato de ser declarativa: o código expressa relações, fatos e leis, e uma pergunta ao programa, uma query, testa relações lógicas dentro dessa rede de inferências que podem ser concluídas logicamente. As respostas do programa são pressões necessárias visto a rede puramente lógica que o precede, tudo mantêm-se coerente entre si. Usa-se a expressividade da lógica de primeira ordem, onde por exemplo, pode-se se explicitar relações entre variáveis que devem ser sempre verdade: “todo

termo X com a propriedade escarlate também possui a propriedade vermelho” e assim por diante.

Foge do escopo uma maior explicação da linguagem, ou descrição do paradigma usado, até por razões práticas, visto que a proposta é um framework, e se um usuário desejar aplicar tal funcionalidade do Prolog para tal sistema, ele próprio deverá identificar a parte dos algoritmos relevante.

Para o presente trabalho foi utilizado o ambiente livre de programação SWI-Prolog, largamente utilizado em pesquisa, educação e aplicações comerciais, além de apresentar comunidade numerosa e ativa. Suas funcionalidades e facilidades de operação facilitaram muito o processo de teste.

5.8 APRESENTAÇÃO DE UM SISTEMA BÁSICO COMPLETO

A função básica do sistema é a seguinte: o objeto o1 possui um sensor de presença publicando no tópico quarto1/presença toda vez que alguém está no quarto1, e o objeto o2 é uma entidade que modela uma fonte de luz que publica no tópico quarto1/luz e subscreve-se no tópico botão1 – com sua interrelação óbvia, a princípio ambos os objetos não sabem da existência do outro, como se estivessem acabados de serem instalados. Os tópicos representam não uma entidade interna ou inerente dos objetos, que nesse caso estão nos microcontroladores das placas ESP8266, e sim um fato sobre o ambiente da casa, como se o termo lógico quarto1/luz estivesse flutuando no próprio centro do quarto.

A Figura 9 incorpora a ideia do framework discutido até aqui – e que será delineada, com suas respectivas explicações a seguir:

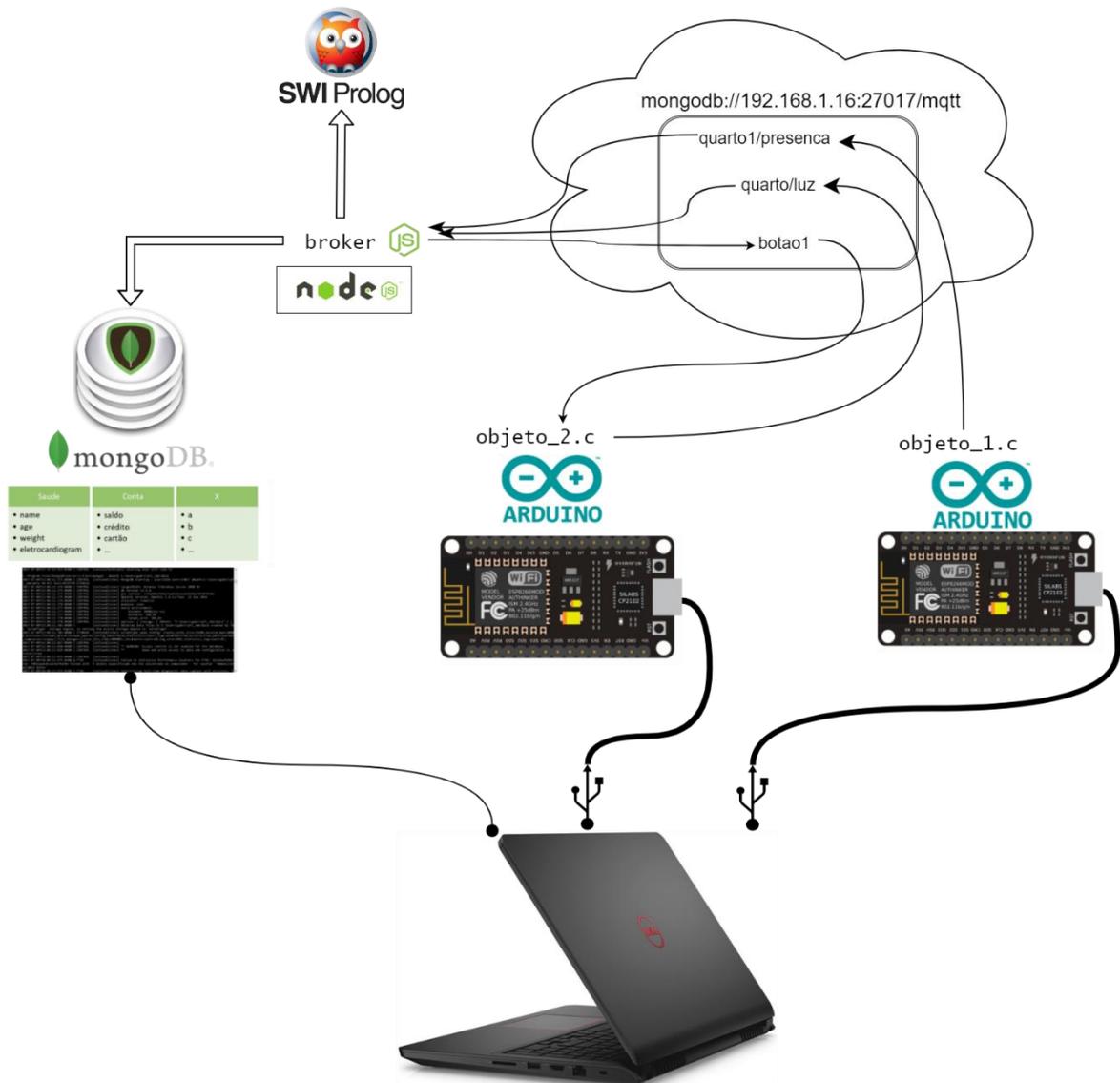


Figura 9 - Esquema do sistema básico completo.

Foram usadas duas unidades ESP8266 (vide figura 9), ambas com código semelhante, a título de modelar dois objetos abstratos. Para esse caso particular um deles, o objeto_1.c da figura contém um código em linguagem c, no qual foi conectado ao Broker (que estoca as informações no banco de dados MongoDB), subscrevendo ao tópico quarto1/presença (se caso chegar algo nesse tópico ele avisa no terminal do NodeMCU via callback), já o objeto_2.c publica em quarto1/luz e se subscreve a botão1. As partes relevantes de ambos são colocados a seguir, sendo que o código completo encontra-se no apêndice:

```
//Pedacos de código relevante do objeto_1.c

#define TOPICO_PUBLISH "quarto1/presença"

void EnviaEstadoOutputMQTT(void)
{
  [...]
}
```

```
//Pedaco de código relevante do objeto_2.c

#define TOPICO_SUBSCRIBE "botao1"
#define TOPICO_PUBLISH "quarto1/luz"

void EnviaEstadoOutputMQTT(void)
{
  if( permission == true)
  {
    [...]
  }

  delay(2000);
}

void mqtt_callback(char* topic, byte* payload, unsigned int length)
{
  [...]
}
```

No Broker, localizado no servidor provido pelo laptop é onde o processo de tratamento de quem se conecta ou desconecta, publica ou se inscreve, acontece. Com isso, qualquer evento de transação pode ser controlado – é a praça, onde todos da cidade dos objetos se encontram. Nesse sistema específico, o broker é conjurado via o Node.js já comentado, ideal para tratamento a eventos, no qual cada evento do sistema é gravado automaticamente (incluindo a exclusão se é o evento oposto) num arquivo chamado systemProlog.pl já, com a sintaxe necessária na forma adequada, pronto para ser inicializado no SWI-Prolog. Alguns pedaços relevantes do código broker.js em Node.js para esses eventos, encontram-se a seguir, sendo que o código completo se encontra no apêndice:

```
// quando cliente se conecta
server.on('clientConnected', function(client) {

  var piece = 'object(' + client.id.toLowerCase() + ').' + '\r\n';
  fs.appendFile('systemProlog.pl', piece, function (err) {
    if (err) throw err;
    console.log('Saved!' + "\r\n");
  });

  console.log('Client connected: ', client.id + "\r\n");
});

// quando cliente se desconecta
server.on('clientDisconnected', function(client) {

  var piece = 'object(' + client.id.toLowerCase() + ').' + '\r\n';
  var n = "";
  fs.readFile('systemProlog.pl', function (err,data) {
    if (err) throw err;
```

```
n = data.toString().replace(piece, "");
console.log(n + "\r\n");

});
```

Acionando o funcionamento do MongoDB, depois rodando o broker.js, já se tem o laptop funcionando como servidor em formato de broker. Por conseguinte, os objetos 1 e 2, modelados acima pelos ESP8266 começam a funcionar interagindo com esse mesmo broker. Pelo código acima isso irá gerar um arquivo systemProlog.pl, em tempo real, correspondente ao esqueleto do sistema – sendo o primeiro passo na extração do conhecimento do sistema. Sendo mais preciso, com os dois objetos funcionando ao mesmo tempo, gerará o seguinte programa em Prolog:

```
object(o1).
object(o2).

topic(quarto1/presenca).
topic(quarto1/luz).
topic(botao1).

sub(o2,botao1).

pub(o1,quarto1/presenca).
pub(o2,quarto1/luz).
```

No SWI-Prolog o programa é compilado, e permanece pronto para ser perguntado sobre alguma característica, como: “? - sub(X, quarto1/presenca)” significando “Que objetos estão subscritos no tópico quarto1/presenca”, etc.

Todavia fazendo jus ao framework inferencial, dever-se-ia conjurar uma inferencial útil que produza conhecimento, descobrindo algo sobre as interconexões do simples sistema que os objetos dentro de seu pequeno mundo não saberiam sem o uso da lógica de primeira ordem. Uma opção clara se mostra, seria bom senso que a luz no quarto1 seja acionada se alguém estiver presente dentro dele, ou seja, “se um objeto X estiver publicando no quarto1/presenca e um objeto Y estiver publicando no quarto1/luz, então o objeto Y subscreverá ao tópico quarto1/presenca (tal que use essa informação”, que em lógica de primeira ordem através do Prolog torna-se:

sub(Y,quarto1/presenca) :- pub(X,quarto1/presenca),pub(Y,quarto1/luz).

Adicionando essa asserção ao programa anterior do Prolog, infere-se que durante todo esse tempo os objetos 1 e 2 estavam “falando” sobre a mesma entidade, e poderiam muito bem serem interligados – via o fato de agora o objeto 2 receber decisões de ligar ou desligar da presença e não somente do botão1. Observa-se que a asserção introduzida produz um contorno de escopo geral (a diferença de algumas cláusulas de generalização da informação dos tópicos), se outro quarto apresentar situação semelhante, a mesma asserção pode ser utilizada, entrelaçando e produzindo conhecimento.

Observa-se que essa última cláusula incorpora exatamente a operação de soma de objetos mostrada pela equação (3.6).

5.9 FRAMEWORK GERAL

Durante a delineação do sistema básico exemplificando uma aplicação foi aprendido como o formato do framework em geral será feito. Ou seja, o resultado da aplicação aumenta a resolução de inspeção conceitual e ajuda a dar o toque final da sequência de tarefas necessárias para produzir qualquer outra aplicação.

Note que será abstraído qualquer uso específico de tecnologias, como MQTT e JavaScript, pois assim se ganha em poder de generalidade e flexibilidade. Por fim a seguinte sequência de tarefas se mostrou:

1. Abstraia os objetos como entidades discretas, como $O_1, O_2, O_3, \dots, O_n$
2. Conecte esses objetos via tópicos de conversa, direcionados, e baseados nos fluxos de informação
3. Desacople as conexões, via uma praça de conversação (como um broker)
4. Com as informações da praça, escreva as informações em asserções em lógica de primeira ordem
5. Realize as inferências coerentes com cada aplicação

6. ESPECULAÇÕES

Certos tópicos por serem não estritamente filosóficos, mas simplesmente por se encontrarem num estado mais bruto de desenvolvimento estão presentes nessa seção. Não significando que o conteúdo ainda implícito não seja valioso ou até talvez esteja controlando e direcionando todo o desenvolver do pensamento do trabalho, tampouco que cada um não mereceria uma seção independente, mas somente há algo mais opaco que pede uma maior atenção e desenvolvimento. Fugidios e um pouco mais selvagens, esses tópicos foram contidos e batalhados somente nessa parte.

Dentre esses desenvolvimentos pertencem aspectos holísticos do que pode ser intuído do desenvolvimento geral da tecnologia IoT como a subseção “Gado Humano”, pontos em cada faceta do desenvolvimento dos quais certas outras linhas de possíveis partidas tomando um rumo independente ficou claro de ser uma possibilidade e induções teóricas originadas na tentativa de observar certos padrões ainda não abordados dentro da literatura a qual o trabalho teve de se imergir.

A razão dessa inclusão é de ser comprometido com o desenvolvimento de todos esses pequenos e as vezes grandes impulsos teóricos rudimentares que permanecem ou nas entrelinhas ou de maneira propositalmente implícita em movimentos como “É fácil ver que...”. Sempre escondidos eles serviriam para talvez esconder de onde está vindo o fluxo de protótipos de ideia, o que seria epistemologicamente antiético e contraproducente para o próprio desenvolvimento do projeto: escrevendo rudimentos de ideias, acaba-se dando vida própria e observa-se características que não se apresentavam lá antes de tê-las descrito.

6.1 ALAVANCA DIGITAL

A figura abaixo é um famoso exemplo de cálculo cinemático realizado pela primeira vez pelo engenheiro mecânico alemão Franz Reuleaux, dito pai da cinemática moderna, onde uma trajetória d deve gerar uma trajetória específica b , e o problema é como e quantas alavancas devem ser conectadas entre tais trajetórias teóricas de tal forma que uma implementação as incorpore corretamente. Resumindo a solução dada por Reuleaux, ela se baseia num centro de rotações instantâneas e

um lugar geométrico dada pelas posições do mecanismo desejada, cujos meandros algorítmicos não serão melhor explicados (REULEAUX, 2012).

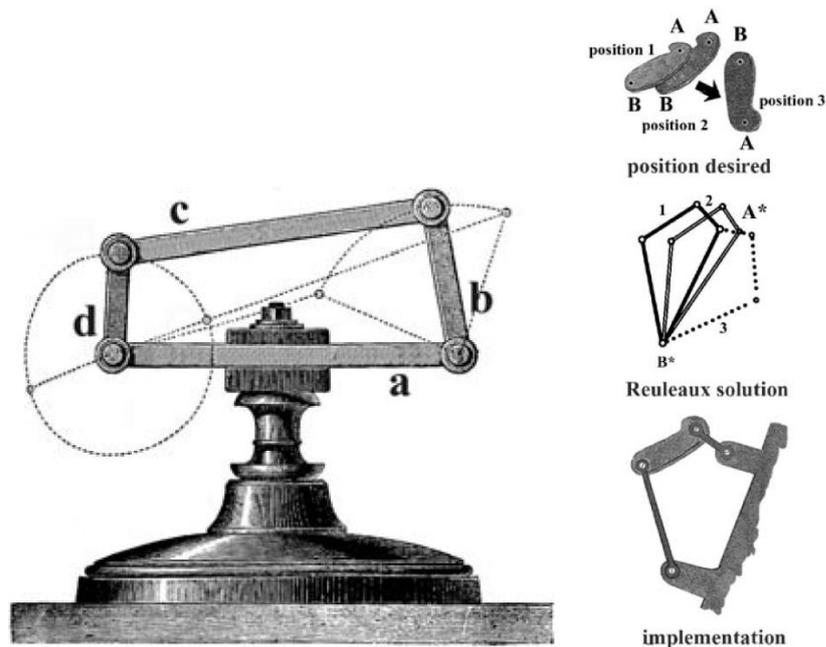


Figura 10 - Alavanca física utilizada por Reuleaux para explicar o seu método, junto com a sua solução.

Fonte: Reuleaux, 2012, sendo a solução de Wilson, 2007.

O importante é o fato de se ter achado um algoritmo geral de um problema de engenharia o qual não necessita de qualquer prática não precisa, que pode se aproximar duma espécie de “arte”. O próprio Reuleaux comentou de como James Watson, o inventor, em seus projetos incipientes utilizava um algoritmo parecido, todavia o método sempre permanecia totalmente implícito. Se alguém pedisse a Watson explicar como ele faz o que faz, ele não saberia explicitar e explicar a sequência de tarefas mentais, provavelmente logo abaixo de seu nível de consciência conceitual, que usava para resolver os seus problemas.

Aqui a especulação mostra-se na sua presença viva: será que existe uma analogia com os sistemas eletrônicos, definindo um certo alvo de controle e um início com certos recursos, no qual algum algoritmo entraria com um nível de abstração preciso e que resolveria explicitamente vários problemas? Colocando de outra maneira: um sistema de objetos conjurados num processo de engenharia reversa entre começo e alvo?

Essa subseção foi incluída dada a razão de se ter percebido que a aplicação do framework em uma aplicação real pode permanecer nebuloso, ou seja, sem uma sequência de tarefas explícita que mencione o que o projetista deve fazer para aplicá-lo, assim emergindo uma distância entre os dois. Sendo que a aplicação permaneceria algo próximo de uma arte, sem um algoritmo explícito. E é exatamente isso que foi localizado, tendo essa seção uma inicial fonte de inspiração para a resolução dessa distância, a saber, do framework abstrato

6.2 GADO HUMANO

Grande fração dos sistemas de objetos que são implementados possuem uma espécie de vontade inerente de cada vez mais se expandir. Uma rede de interligações, talvez até pela sua dinâmica interna, beira uma incompletude ontológica – falta algo para se tornar inteiramente real. Óbvio que o sistema não pensa, não propõe de maneira explícita que “deseja mais objetos” ou suplica “por favor, estou precisando de um sensor de movimento no quarto de número 2”. Esse ímpeto de aglomeração é puramente imanente aos movimentos de design do projetista e do uso daqueles que operam a vivência dele. Em outras palavras, ambos se deparam com pequenas perguntas de adição de tal sensor ou microcontrolador nas funções presentes, mesmo esses não sendo previstos inicialmente do protótipo do projeto.

O princípio de norteamiento é que quanto mais informação compartilhada entre os usuários em um contexto qualquer, e eles tendo objetivos coerentes, melhor para todos. Portanto, para fins de controle mais é melhor.

No tópico das casas inteligentes – a automação residencial já comentada – não há razão alguma contra a progressiva aglomeração de mais cômodos na rede, ou sua expansão de sensores, a menos por razões de brechas indesejadas de segurança ou maior complexidade. Todavia, é baixa o aumento da complexidade em contrapartida com a obtenção de mais funcionalidade. Persuadindo que se postule a tese que essa situação tende a se repetir, a saber o adição progressivo de objetos, com grande probabilidade, e sem que esteja pré-estabelecido no esquema inicial do projeto. O controle é proliferador e autogerador de sua própria condição de aumento.

E o que isso tem com relação ao título dessa seção? Para essa resposta, antes é bom que se inspecione a seguinte passagem inocente numa referência sobre

Internet das Coisas no contexto de apresentação de suas potencialidades onde RTLS é uma sigla, no inglês, significando sistemas de localização em tempo real:

A good example of RTLS and its role in the Internet of Things is visible at Oregon Health Sciences University (OHSU). The Portland, Oregon, health and research institution tags assets ranging from infusion pumps to crutches so that they are easy to locate. In addition they are able to track performance data related to the device. This approach not only saves time that would otherwise be spent hunting down equipment, it helps ensure that devices are in working order. OHSU is now looking into tagging patients and clinicians to better understand where they spend time, how they move around within the facility, and how long patients wait in a room before a clinician arrives. (GREENGARD, 2015)

O leitor atencioso provavelmente notou algo de suspeito na última frase: a cogitação de rastrear os médicos e pacientes para “entender” onde gastam o seu tempo, sua localização e demora em cada locação, é uma fonte – depois dum mínimo de reflexão crítica – de ultra controle gerencial sobre os sujeitos. É claro que os vigiados teoricamente concordariam por configuração padrão das relações trabalhistas, e poder-se-ia dizer que todo o uso seria benigno até para os próprios médicos e pacientes.

Todavia, quem controla o sistema que controla? A gerência a princípio, a qual nunca permanentemente é uma pessoa só, funcionando como uma mini instituição, portanto inculcando a atenção na estrutura posta como mecanismo de vigilância, e não simplesmente quem estaria controlando em tal momento – o importante é a presença empírica do sistema funcionando legalmente. Se quem usa num determinado momento respeita a legalidade não é crucial, adormece a constante chance de ser abusado, ou pior: o abuso ganhar soberania e se tornar legal com o tempo.

Esse ponto crítico, embora sendo apenas uma instância, é uma tendência geral. Pressionando o desenvolver desse raciocínio numa metáfora – vide o título – dum gado, só que dessa vez, humano. E aqui suplica-se a imaginação do leitor na conjuração de todo aparato que existe para controle de um gado numa entidade moderna, com todas as suas etiquetas auriculares, baias de separação, banco de dados com todas as informações da saúde (afinal exatamente como no caso do paciente, uma vaca morta é uma fonte de recurso a menos). Não é só somente uma coincidência temática, puramente imagética, como na observação longínqua dum grupo grande de pessoas com um formigueiro – vai muito além. A racionalidade, o ímpeto, começando com boas intenções, que é exatamente o que se replica: uma vez uma estrutura qualquer de tratamento humano como gado é implementada – como

viu-se a tese no início dessa seção – ela tenderá a crescer e produzir sua própria condição de proliferação.

Abstrações são extremamente úteis, todavia sua própria razão de ser é o seu perigo: retirando diversas informações a princípio não relevantes, cria-se um meio opaco de observação, e se esquece que do outro lado, acreditem se quiser, também podem existir sujeitos.

7. CONCLUSÃO

Obeve-se um sistema básico, ou melhor, um framework de Internet das Coisas embasado em fundamentos filosóficos bem delineados (pós-fenomenologia), o que tanto não se encontrou na literatura e em implementações pertinentes, quanto se apresentou uma solução ao problema da falta de melhor definição da semelhança de família desses sistemas. Além disso foi utilizado um grau de abstração, usando a linguagem de objetos abstratos de uso geral como unidades básicas (O_1 , O_2 , etc.) e suas respectivas conversas via tópicos no MQTT, e por conseguinte o entrelaçamento semântico com possíveis inferências sobre a estrutura do sistema de objetos via a linguagem de programação lógica Prolog.

Uma vez montado o processo de tradução constante das informações do broker em um código em Prolog – utilizando um programa em Node.js, no texto chamado de `controller.js` –, as possibilidades são imensas, quaisquer funções que retirem informações da rede podem ser injetadas de maneira extremamente rápida (apenas um comando `make` no Prolog). Dependendo apenas quais serão os interesses do implementador em tal contexto – daí a abstração multe uso do framework – a programação lógica pode ser maleada para essa exata aplicação: seja uma rede que pressione a otimização de tempo dentre as conexões dos objetos, então uma função filtradora no Prolog seria um algoritmo de grafos para o menos caminho, por meio do qual perguntas no estilo “? – `caminho_mais_curto(X,Y)`” podem ser respondidas.

Essa montagem de programas lógicos para explorar e criar conhecimento no sistema é de imensa generalização e uso, só dependendo da criatividade e problemas enfrentados pelo engenheiro. As portas estão abertas uma vez que o chão do framework foi montado. Em outras palavras pode-se explicitar os seguintes passos:

6. Abstraia os objetos como entidades discretas, como $O_1, O_2, O_3, \dots, O_n$
7. Conecte esses objetos via tópicos de conversa, direcionados, e baseados nos fluxos de informação
8. Desacople as conexões, via uma praça de conversação (como um broker)
9. Com as informações da praça, escreva as informações em asserções em lógica de primeira ordem
10. Realize as inferências coerentes com cada aplicação

É importante frisar uma meta-conclusão, a saber um movimento de interpretação não do trabalho em si, mas da sua forma depois de esvaziada de conteúdo, aparece como aprendizado totalmente não previsto pelo desenrolar inicial do projeto. Tanta movimentação conceitual e fertilização – como foi chamada essa exploração pela filosofia da tecnologia – causou uma razão de ser dos passos feitos que ultrapassam o contexto da própria Internet das Coisas sistêmica, e não só pelo fato de ter-se transformado num framework mais geral. O sentido é que alguns empreendimentos – nesse caso a princípio somente em engenharia – necessitam uma mínima reflexão sobre esse ímpeto inicial muitas vezes cego. Pelo menos algumas perguntas como:

- “Que conceito estamos usando para começar a conversar sobre a solução deste problema? ” – Viu-se o quão necessário foi realizar uma mínima análise do conceito de “coisa”/”*things*” o qual aparece como uma espécie de coletivo ou conjunto. Todavia acaba por eclipsar diversas aplicações ou até realmente ajudar com um “empurrãozinho” para uma abstração – no nosso caso a linguagem O1, etc. Sendo benção ou maldição é melhor refletir sobre as ferramentas conceituais antes.
- “A pergunta foi posta de maneira geral/particular? ” – Medir o quanto de particularidade – aqui definido como nível de aplicabilidade singular – uma pergunta tem modula praticamente todo o desenvolvimento da história da solução. O pulo de “Automação residencial usando Internet das Coisas” para o framework mais geral foi fruto desse gênero de questionamento.
- “O que é comum ‘entre’ e não ‘sobre’? ” – Compara-se diversas instâncias de sistemas de IoT, é só o fato de existirem coisas conectadas entre si e na Internet que faz delas serem dessa estranha categoria IoT? E se não for somente isso, e existirem outras propriedades implícitas como emergência de algo qualitativamente novo assim que os sistemas ficam muito complexos ou se interligam? – Utilizando essas disposições é claro que o trabalho tentou abordar cada uma dessas perguntas, checar a seção 3.3 e partes da seção 5 para maiores delineações e desenvolvimento.

Para trabalhos futuros podem-se citar os seguintes tópicos e zonas de ação:

1. Automatizar de alguma maneira o uso das inferências do Prolog

dentro do sistema, de forma adaptativa e automática via o processo controlando o broker com o próprio comunicando as mudanças para os objetos. Criando um algoritmo de aprendizagem sistêmica.

2. Inserir uma lógica mais expressiva dentro dos recursos do Prolog.
3. Modelar matematicamente as redes dos clientes do *broker*, e apresentar visualmente os grafos representantes via uma dashboard em browser.

8. REFERÊNCIAS

ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, v. 54, p. 2787-2805, 2010.

BRANDOM, R. **Articulating reasons**. [S.l.]: Harvard University Press, 2009.

BREY, P. Technology as extension of human faculties, 2000.

CHODOROW, K. **MongoDB: The Definitive Guide: Powerful and Scalable Data Storage**. [S.l.]: "O'Reilly Media, Inc.", 2013.

FLANAGAN, D. **JavaScript: the definitive guide**. [S.l.]: "O'Reilly Media, Inc.", 2006.

GREENGARD, S. **The internet of things**. [S.l.]: MIT Press, 2015.

GUBBI, J. et al. Internet of Things (IoT): A vision, architectural elements, and future directions. **Future generation computer systems**, v. 29, p. 1645-1660, 2013.

HOLLER, J. et al. **From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence**. [S.l.]: Academic Press, 2014.

HUNKELER, U.; TRUONG, H. L.; STANFORD-CLARK, A. **MQTT-S A publish/subscribe protocol for Wireless Sensor Networks**. Communication systems software and middleware and workshops, 2008. comsware 2008. 3rd international conference on. [S.l.]: [s.n.]. 2008. p. 791-798.

KAPP, E. **Grundlinien einer Philosophie der Technik: zur Entstehungsgeschichte der Kultur aus neuen Gesichtspunkten**. [S.l.]: Meiner Verlag, v. 675, 2015.

LADYMAN, J. et al. **Every thing must go: Metaphysics naturalized**. [S.l.]: Oxford University Press on Demand, 2007.

MANDUCA, C. A.; MOGK, D. W.; OTHERS. **Earth and mind: How geologists think and learn about the earth**. [S.l.]: Geological Society of America, v. 413, 2006.

MCLUHAN, M. **Understanding media: The extensions of man**. [S.l.]: MIT press, 1994.

RAYES, A.; SAMER, S. **Internet of Things—From Hype to Reality**. [S.l.]: Springer, 2017.

REULEAUX, F.; FERGUSON, E. S. **Kinematics of machinery: outlines of a theory of machines**. [S.l.]: Courier Corporation, 2012.

WILSON, M. **Wandering significance: An essay on conceptual behavior**. [S.l.]: Oxford University Press, 2008.

ZIOUVELOU, X. et al. Crowd-Driven IoT/loE Ecosystems: A Multidimensional Approach. In: _____ **Beyond the Internet of Things**. [S.l.]: Springer, 2017. p. 341-375.

9. APÊNDICES

9.1 CÓDIGO DO objeto_1.c PROGRAMADO EM C:

```

#include <ESP8266WiFi.h> // Importa a Biblioteca ESP8266WiFi
#include <PubSubClient.h> // Importa a Biblioteca PubSubClient

#include <time.h>
#include <stdlib.h>
#include <string>
#include <sstream>
namespace patch
{
    template < typename T > std::string to_string( const T& n )
    {
        std::ostringstream stm ;
        stm << n ;
        return stm.str() ;
    }
}

//defines:
//defines de id mqtt e tópicos para publicação e subscribe

#define TOPICO_PUBLISH "quarto1/presenca"

#define ID_MQTT "o1" //id mqtt (para identificação de sessão)
                    //IMPORTANTE: este deve ser único no broker (ou seja,
                    // se um client MQTT tentar entrar com o mesmo
                    // id de outro já conectado ao broker, o broker
                    // irá fechar a conexão de um deles).

//defines - mapeamento de pinos do NodeMCU
#define D0 16
#define D1 5
#define D2 4
#define D3 0
#define D4 2
#define D5 14
#define D6 12
#define D7 13
#define D8 15
#define D9 3
#define D10 1

```

```

// WIFI
const char* SSID = "AndroidAP"; // SSID / nome da rede WI-FI que deseja se conectar
const char* PASSWORD = "oyfv2113"; // Senha da rede WI-FI que deseja se conectar

// MQTT
const char* BROKER_MQTT = "iot.eclipse.org"; //URL do broker MQTT que se deseja
utilizar
//const char* BROKER_MQTT = "mqtt://192.168.1.16";
//IPAddress BROKER_MQTT(192, 168, 1, 16);

int BROKER_PORT = 1883; // Porta do Broker MQTT

//Variáveis e objetos globais
WiFiClient espClient; // Cria o objeto espClient
PubSubClient MQTT(espClient); // Instancia o Cliente MQTT passando o objeto espClient
char EstadoSaida = '0'; //variável que armazena o estado atual da saída

//Prototypes
void InitOutput(void);
void initSerial();
void initWiFi();
void initMQTT();
void blink_red();
void reconnectWiFi();
void mqtt_callback(char* topic, byte* payload, unsigned int length);
void VerificaConexoesWiFiMQTT(void);

/*
 * Implementações das funções
 */
void setup()
{
  //inicializações:
  InitOutput();
  initSerial();
  initWiFi();
  initMQTT();
}

void InitOutput(void)
{
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH);

  pinMode(D0, OUTPUT);
  digitalWrite(D0, HIGH);
}

void initSerial()

```

```

{
  Serial.begin(115200);
}

void initWiFi()
{
  delay(10);
  Serial.println("-----Conexao WI-FI-----");
  Serial.print("Conectando-se na rede: ");
  Serial.println(SSID);
  Serial.println("Aguarde");

  reconnectWiFi();
}

//Função: inicializa parâmetros de conexão MQTT(endereço do
// broker, porta e seta função de callback)
//Parâmetros: nenhum
//Retorno: nenhum
void initMQTT()
{
  MQTT.setServer(BROKER_MQTT, BROKER_PORT); //informa qual broker e porta deve
ser conectado
  MQTT.setCallback(mqtt_callback); //atribui função de callback (função chamada
quando qualquer informação de um dos tópicos subscritos chega)
}

void blink_red()
{
  digitalWrite(LED_BUILTIN, LOW);
  delay(50);
  digitalWrite(LED_BUILTIN, HIGH);
}

int permission = 0;
//PRINCIPAL PARTE DO CÓDIGO
void mqtt_callback(char* topic, byte* payload, unsigned int length)
{
  String msg;

  //obtem a string do payload recebido
  for(int i = 0; i < length; i++)
  {
    char c = (char)payload[i];
    msg += c;
  }

  Serial.println();
  Serial.print("Action from actions received, under the topic name of: ");
  Serial.print(topic);
  Serial.println();
  Serial.print("Content from action is: ");
  Serial.print(msg);
}

```

```

Serial.println();
Serial.println();
if( msg.equals("aceitar"))
{
    permission = true;
}

    blink_red();
}

//Função: reconecta-se ao broker MQTT (caso ainda não esteja conectado ou em caso de a
conexão cair)
//    em caso de sucesso na conexão ou reconexão, o subscribe dos tópicos é refeito.
//Parâmetros: nenhum
//Retorno: nenhum
void reconnectMQTT()
{
    while (!MQTT.connected())
    {
        Serial.print("* Tentando se conectar ao Broker MQTT: ");
        Serial.println(BROKER_MQTT);
        if (MQTT.connect(ID_MQTT))
        {
            Serial.println("Conectado com sucesso ao broker MQTT!");
            MQTT.subscribe("actions");
            MQTT.publish("objeto/esteira", "true");
            blink_red();
            Serial.println("Published at topico objeto/esteira.");
        }
        else
        {
            Serial.println("Falha ao reconectar no broker.");
            Serial.println("Havera nova tentatica de conexao em 2s");
            delay(2000);
        }
    }
}

//Função: reconecta-se ao WiFi
//Parâmetros: nenhum
//Retorno: nenhum
void reconnectWiFi()
{
    //se já está conectado a rede WI-FI, nada é feito.
    //Caso contrário, são efetuadas tentativas de conexão
    if (WiFi.status() == WL_CONNECTED)
        return;

    WiFi.mode(WIFI_STA);
    WiFi.begin(SSID, PASSWORD); // Conecta na rede WI-FI

    while (WiFi.status() != WL_CONNECTED)
    {

```

```

        delay(100);
        Serial.print(".");
    }

    Serial.println();
    Serial.print("Conectado com sucesso na rede ");
    Serial.print(SSID);
    Serial.println("IP obtido: ");
    Serial.println(WiFi.localIP());
}

//Função: verifica o estado das conexões WiFi e ao broker MQTT.
//    Em caso de desconexão (qualquer uma das duas), a conexão
//    é refeita.
//Parâmetros: nenhum
//Retorno: nenhum
void VerificaConexoesWiFiEMQTT(void)
{
    if (!MQTT.connected())
        reconnectMQTT(); //se não há conexão com o Broker, a conexão é refeita
    MQTT.subscribe("actions");
    reconnectWiFi(); //se não há conexão com o WiFi, a conexão é refeita
}

void EnviaEstadoOutputMQTT(void)
{
    if( permission == true)
    {
        MQTT.publish("saude/electrocardiogram","70" );
        Serial.println("Published at topico saude/electrocardiogram.");
        blink_red();
    }

    delay(2000);
}

//programa principal
void loop()
{
    //garante funcionamento das conexões WiFi e ao broker MQTT
    VerificaConexoesWiFiEMQTT();

    //envia o status de todos os outputs para o Broker no protocolo esperado
    EnviaEstadoOutputMQTT();

    //keep-alive da comunicação com broker MQTT
    MQTT.loop();
}

```

9.2 CÓDIGO DO objeto_2.c PROGRAMADO EM C:

```

#include <ESP8266WiFi.h> // Importa a Biblioteca ESP8266WiFi
#include <PubSubClient.h> // Importa a Biblioteca PubSubClient

#include <time.h>
#include <stdlib.h>
#include <string>
#include <sstream>
namespace patch
{
    template < typename T > std::string to_string( const T& n )
    {
        std::ostringstream stm ;
        stm << n ;
        return stm.str() ;
    }
}

//defines:
//defines de id mqtt e tópicos para publicação e subscribe
#define TOPICO_SUBSCRIBE "botao1"
#define TOPICO_PUBLISH "quarto1/luz"

#define ID_MQTT "o2" //id mqtt (para identificação de sessão)
                    //IMPORTANTE: este deve ser único no broker (ou seja,
                    // se um client MQTT tentar entrar com o mesmo
                    // id de outro já conectado ao broker, o broker
                    // irá fechar a conexão de um deles).

//defines - mapeamento de pinos do NodeMCU
#define D0 16
#define D1 5
#define D2 4
#define D3 0
#define D4 2
#define D5 14
#define D6 12
#define D7 13
#define D8 15
#define D9 3
#define D10 1

// WIFI
const char* SSID = "AndroidAP"; // SSID / nome da rede WI-FI que deseja se conectar
const char* PASSWORD = "oyfv2113"; // Senha da rede WI-FI que deseja se conectar

```

```

// MQTT
const char* BROKER_MQTT = "iot.eclipse.org"; //URL do broker MQTT que se deseja
utilizar
//const char* BROKER_MQTT = "mqtt://192.168.1.16";
//IPAddress BROKER_MQTT(192, 168, 1, 16);

int BROKER_PORT = 1883; // Porta do Broker MQTT

//Variáveis e objetos globais
WiFiClient espClient; // Cria o objeto espClient
PubSubClient MQTT(espClient); // Instancia o Cliente MQTT passando o objeto espClient
char EstadoSaida = '0'; //variável que armazena o estado atual da saída

//Prototypes
void InitOutput(void);
void initSerial();
void initWiFi();
void initMQTT();
void blink_red();
void reconnectWiFi();
void mqtt_callback(char* topic, byte* payload, unsigned int length);
void VerificaConexoesWiFiMQTT(void);

/*
 * Implementações das funções
 */
void setup()
{
  //inicializações:
  InitOutput();
  initSerial();
  initWiFi();
  initMQTT();
}

void InitOutput(void)
{
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, HIGH);

  pinMode(D0, OUTPUT);
  digitalWrite(D0, HIGH);
}

void initSerial()
{
  Serial.begin(115200);
}

```

```

void initWiFi()
{
    delay(10);
    Serial.println("-----Conexao WI-FI-----");
    Serial.print("Conectando-se na rede: ");
    Serial.println(SSID);
    Serial.println("Aguarde");

    reconnectWiFi();
}

//Função: inicializa parâmetros de conexão MQTT(endereço do
// broker, porta e seta função de callback)
//Parâmetros: nenhum
//Retorno: nenhum
void initMQTT()
{
    MQTT.setServer(BROKER_MQTT, BROKER_PORT); //informa qual broker e porta deve
ser conectado
    MQTT.setCallback(mqtt_callback); //atribui função de callback (função chamada
quando qualquer informação de um dos tópicos subscritos chega)
}

void blink_red()
{
    digitalWrite(LED_BUILTIN, LOW);
    delay(50);
    digitalWrite(LED_BUILTIN, HIGH);
}

int permission = 0;
//PRINCIPAL PARTE DO CÓDIGO
void mqtt_callback(char* topic, byte* payload, unsigned int length)
{
    String msg;

    //obtem a string do payload recebido
    for(int i = 0; i < length; i++)
    {
        char c = (char)payload[i];
        msg += c;
    }

    Serial.println();
    Serial.print("Action from actions received, under the topic name of: ");
    Serial.print(topic);
    Serial.println();
    Serial.print("Content from action is: ");
    Serial.print(msg);
    Serial.println();
    Serial.println();
    if( msg.equals("aceitar"))
    {

```

```

    permission = true;
}

    blink_red();
}

//Função: reconecta-se ao broker MQTT (caso ainda não esteja conectado ou em caso de a
conexão cair)
//    em caso de sucesso na conexão ou reconexão, o subscribe dos tópicos é feito.
//Parâmetros: nenhum
//Retorno: nenhum
void reconnectMQTT()
{
    while (!MQTT.connected())
    {
        Serial.print("* Tentando se conectar ao Broker MQTT: ");
        Serial.println(BROKER_MQTT);
        if (MQTT.connect(ID_MQTT))
        {
            Serial.println("Conectado com sucesso ao broker MQTT!");
            MQTT.subscribe("actions");
            MQTT.publish("objeto/esteira", "true");
            blink_red();
            Serial.println("Published at topico objeto/esteira.");
        }
        else
        {
            Serial.println("Falha ao reconectar no broker.");
            Serial.println("Havera nova tentatica de conexao em 2s");
            delay(2000);
        }
    }
}

//Função: reconecta-se ao WiFi
//Parâmetros: nenhum
//Retorno: nenhum
void reconnectWiFi()
{
    //se já está conectado a rede WI-FI, nada é feito.
    //Caso contrário, são efetuadas tentativas de conexão
    if (WiFi.status() == WL_CONNECTED)
        return;

    WiFi.mode(WIFI_STA);
    WiFi.begin(SSID, PASSWORD); // Conecta na rede WI-FI

    while (WiFi.status() != WL_CONNECTED)
    {
        delay(100);
        Serial.print(".");
    }
}

```

```
Serial.println();
Serial.print("Conectado com sucesso na rede ");
Serial.print(SSID);
Serial.println("IP obtido: ");
Serial.println(WiFi.localIP());
}

//Função: verifica o estado das conexões WiFi e ao broker MQTT.
// Em caso de desconexão (qualquer uma das duas), a conexão
// é refeita.
//Parâmetros: nenhum
//Retorno: nenhum
void VerificaConexoesWiFiMQTT(void)
{
    if (!MQTT.connected())
        reconnectMQTT(); //se não há conexão com o Broker, a conexão é refeita
    MQTT.subscribe("actions");
    reconnectWiFi(); //se não há conexão com o WiFi, a conexão é refeita
}

void EnviaEstadoOutputMQTT(void)
{
    if( permission == true)
    {
        MQTT.publish("saude/electrocardiogram","70" );
        Serial.println("Published at topico saude/electrocardiogram.");
        blink_red();
    }

    delay(2000);
}

//programa principal
void loop()
{
    //garante funcionamento das conexões WiFi e ao broker MQTT
    VerificaConexoesWiFiMQTT();

    //envia o status de todos os outputs para o Broker no protocolo esperado
    EnviaEstadoOutputMQTT();

    //keep-alive da comunicação com broker MQTT
    MQTT.loop();
}
```

9.3 CÓDIGO DO *BROKER.JS* PROGRAMADO EM NODE.JS:

```

var mosca = require('mosca')
var url = "mongodb://192.168.1.16:27017/mqtt";
var fs = require('fs');

var ascoltatore = {
//using ascoltatore
type: 'mongo',
url: 'mongodb://192.168.1.16:27017/mqtt',
pubsubCollection: 'ascoltatori',
mongo: {}
};

var moscaSettings = {
port: 1883,
backend: ascoltatore,
persistence: {
factory: mosca.persistence.Mongo,
url: 'mongodb://192.168.1.16:27017/mqtt'
},
stats: false,
host: 'localhost'
};

var server = new mosca.Server(moscaSettings);
server.on('ready', setup);

var MongoClient = require('mongodb').MongoClient;
/*

updatePrologBridge()
function updatePrologBridge() {
setTimeout(() => {
}, 5000)

console.log('Updating Prolog Bridge')
MongoClient.connect(url, function(err, db) {
if (err) throw err;
var newvalues = { electrocardiogram: message.toString() };
db.collection("customers").updateOne(myquery, newvalues, function(err, res) {
if (err) throw err;
console.log("1 record updated");
db.close();
});
});
}
*/
// fired when a client connects
server.on('clientConnected', function(client) {

var piece = 'object(' + client.id.toLowerCase() + ').' + '\r\n';
fs.appendFile('systemProlog.pl', piece, function (err) {
if (err) throw err;
console.log('Saved!' + "\r\n");
});

        console.log('Client connected: ', client.id + "\r\n");
});

```

```

// fired when a client disconnects
server.on('clientDisconnected', function(client) {

var piece = 'object(' + client.id.toLowerCase() + ').'+ '\r\n';
var n = "";
fs.readFile('systemProlog.pl', function (err,data) {
if (err) throw err;

n = data.toString().replace(piece,"");
console.log(n + "\r\n");

});

fs.writeFile('systemProlog.pl', n, function (err) {
console.log('ENTROU'+ "\r\n");
if (err) throw err;
});
console.log('Client Disconnected: ', client.id + "\r\n");
});

// fired when a message is received
server.on('published', function(packet, client) {

if(client != undefined)
{
var piece = 'pub(' + client.id.toLowerCase() + ', ' + packet.topic + ').'+ '\r\n';

console.log('Client', client.id);
console.log('Published', packet.payload + "\r\n");

fs.appendFile('systemProlog.pl', piece , function (err) {
if (err) throw err;
console.log('Saved!' + "\r\n");

});
}

});

server.on('subscribed', function(topic, client) {

var piece = 'sub(' + client.id.toLowerCase() + ', ' + topic + ').'+ '\r\n';

fs.appendFile('systemProlog.pl', piece, function (err) {
if (err) throw err;
console.log('Saved!' + "\r\n");
});

console.log('Client', client.id );
console.log('Subscribed to ', topic + "\r\n");

});

server.on('unsubscribed', function(topic, client) {

var piece = 'sub(' + client.id.toLowerCase() + ', ' + topic + ').'+ '\r\n';
var n = "";
fs.readFile('systemProlog.pl', function (err,data) {

```

```
if (err) throw err;

n = data.toString().replace(piece,"");

});

fs.writeFile('systemProlog.pl', n, function (err) {
if (err) throw err;
});
console.log('Client', client.id );
console.log('Unsubscribed from ', topic + "\r\n");

});

// fired when the mqtt server is ready
function setup() {
console.log('Mosca server is up and running' + '\r\n')
}

process.stdin.resume();//so the program will not close instantly

function exitHandler(options, err) {
if (options.cleanup) console.log('clean');
if (err) console.log(err.stack);
if (options.exit)
{
fs.unlink('systemProlog.pl', function (err) {
if (err) throw err;
console.log('File deleted!');
});
process.exit();
}
}

//do something when app is closing
process.on('exit', exitHandler.bind(null,{cleanup:true}));

//catches ctrl+c event
process.on('SIGINT', exitHandler.bind(null, {exit:true}));

//catches uncaught exceptions
process.on('uncaughtException', exitHandler.bind(null, {exit:true}));
```