

Um ambiente de geração massiva de dispositivos Virtuais em Internet das Coisas

Saymon Castro de Souza*, José Gonçalves P. Filho[†], Lucas G. de P. Salgado[‡] e Rafael B. Pedruzzi[§]

*Instituto Federal do Espírito Santo, Brazil

^{†‡§}Universidade Federal do Espírito Santo, Brazil

Email: *saymon@ifes.edu.br, [†]zegonc@inf.ufes.br, [‡]lucas.gi.salgado@gmail.com, [§]rafaelbelmockpedruzzi@hotmail.com

Resumo—Recentemente, pesquisas em IoT tem sido direcionadas a novos desafios relacionados a um cenário mais amplo de disponibilidade massiva de dispositivos. No entanto, a concretização desse cenário massivo em um ambiente real é onerosa e algumas vezes inviável. Neste trabalho propomos um ambiente de geração massiva de dispositivos virtuais de Internet das Coisas que uniformiza o processo de interação e a estratégia de descrição dos dispositivos virtuais, adotando a recomendação W3C Web of Things. Para validar a proposta, foi conduzido um experimento que gerou um conjunto massivo de dispositivos virtuais e avaliou o tempo de execução e a funcionalidade dos dispositivos gerados.

Index Terms—Massive Internet of Things, Web of Things, Simulator, Ontology.

I. INTRODUÇÃO

A Internet das Coisas está transformando a visão padrão da Web de um conjunto de documentos e links digitais em uma Internet totalmente integrada, que inclui dispositivos físicos, bem como elementos cibernéticos. Essa nova Web, combinando os mundos cibernético e físico, cria um novo ecossistema com novas oportunidades de programação por meio das interações e interconexões dos dois domínios.

A integração do ecossistema de dispositivos inteligentes requer um esforço considerável e limita as oportunidades de programação para espaços inteligentes. Um sensor por si só tem valor limitado ao domínio de aplicação ao qual ele foi projetado; entretanto, existem enormes oportunidades para mercados abertos de serviços que combinam sensores, atuadores e múltiplas fontes de informação. Segundo [1], este desafio introduz, como primeiro passo, um requisito para uma maneira uniforme de descrever as coisas em espaços inteligentes em termos do que é uma coisa (por exemplo, seus componentes), o que faz (por exemplo, os serviços oferecidos) e como ela se comunica (por exemplo, quais protocolos são suportados). Essa descrição abre um caminho para resolver os desafios de integração, configuração e gerenciamento de Coisas, além de permitir interações de forma intuitiva. Para fazer isso, as interfaces lógicas das Coisas são usadas para apresentar os recursos e serviços do dispositivo usando linguagens e anotações da Web Semântica, como DDL (*Device Description Language*) [2], IoT-DDL [1], CoRE-TD [3] e WoT-TD [4]. Em particular, a iniciativa do W3C Web of Things [4] tem se destacado como uma abordagem promissora, embora ainda pouco explorada na literatura em termos de implementação e estudos de caso. Outra importante discussão nessa direção,

destacado em [5], [6], [7], se refere à relevância de Ontologias na descrição semântica dos dispositivos IoT e dos dados gerados a partir deles.

Recentemente, [8] inaugurou uma discussão a respeito dos desafios e estratégias de como aplicações podem atuar em um cenário de Internet das Coisas massivos. A concretização de tal cenário pressupõe a existência de uma quantidade imensa de dispositivos IoT, que vão demandar uma certa uniformidade no que se refere à maneira de interagir com um particular dispositivo. No entanto, a concepção da IoT massiva em um ambiente real com vistas a explorar os seus benefícios e superar os desafios de pesquisa inerentes à área, pode se mostrar inviável em termos de pronta disponibilidade de dispositivos físicos. Por exemplo, a criação de protótipos em larga escala usando um número gigantesco de plataformas de hardware distintas pode não ser prático durante a fase inicial do projeto e sua avaliação exploratória em função de restrições econômicas e operacionais, especialmente quando a confiabilidade e a utilidade da proposta em consideração ainda carecem de uma melhor avaliação. Além disso, a criação de experimentos confiáveis e reproduzíveis envolvendo hardware real pode ser um desafio e muitas vezes requer experiência específica e conhecimento de domínio [9].

Assim, um ciclo típico do processo de pesquisa de IoT, que começa com a formulação de uma ideia e culmina com a implantação no mundo real, compreende elementos virtuais e reais [10]. Uma prova de conceito geralmente é realizada no domínio virtual usando um ambiente de simulação e o protótipo real subsequente é desenvolvido por meio de experimentação em um testbed.

Este artigo apresenta uma solução para a geração massiva de dispositivos IoT virtuais em conformidade com a recomendação W3C Web of Things [4], que padroniza a forma de expor as características e as funcionalidades de um dispositivo IoT, incluindo a forma de interação, os protocolos de comunicação, os formatos de dados, e a descrição semântica dos seus componentes, dentre outros.

Para validar a arquitetura proposta foi gerado um conjunto extenso de dispositivos IoT virtuais, com diversidade de interações e uso de ontologias para definir conceitualmente cada componente do dispositivo virtual. Os dispositivos virtuais gerados foram hospedados em uma plataforma de nuvem computacional e, por fim, foi demonstrada a funcionalidade de um dispositivo particular, explorando suas interações e

desempenho.

II. O AMBIENTE DE IoT MASSIVA

A realização plena da Internet das Coisas e a sua adoção como uma infraestrutura de massa, acessível a todos os setores da sociedade, impõe a superação de uma série de desafios tecnológicos. A escala apresentada pelos ambientes IoT massivos eleva estes desafios a um outro patamar, exigindo preocupações adicionais com eficiência operacional, confiabilidade e disponibilidade. Todavia, como destacado, reproduzir o cenário real com vistas aos testes das aplicações IoT massivas, pode ser uma tarefa árdua e custosa, dada a diversidade e complexidade do ambiente concreto de operação.

Projetos de implantação de cidades inteligentes são exemplos de iniciativas que dão a dimensão do enorme esforço e investimento que são necessários para a provisão massiva de dispositivos físicos inteligentes e a implantação da infraestrutura requerida para a sua operação plena. Uma amostra nesta direção é o projeto SmartSantander [11]. Trata-se de um projeto de pesquisa que propõe uma plataforma de experimentação única em escala mundial, suportando aplicações e serviços típicos para uma cidade inteligente, incluindo pesquisas experimentais avançadas sobre tecnologias IoT. Neste projeto foram implantados mais de 20.000 nós sensores, com um orçamento total de mais de oito milhões de euros.

Um outro exemplo no mesmo domínio é o projeto CityPulse [12]. O projeto foi realizado com a meta ambiciosa de coletar mais de uma centena de cenários de potenciais aplicações IoT visando obter a maior diversidade possível de ideias para discussão com as partes interessadas da cidade e com a comunidade em geral. O levantamento realizado no projeto revelou a dimensão dos eventuais novos negócios que podem emergir do usufruto de uma infraestrutura existente de Internet das Coisas massiva. Foi também constatado pelo projeto que o desenvolvimento de potenciais novos negócios está limitado pela disponibilidade de infraestrutura massiva de dispositivos IoT. Os resultados colhidos demonstraram a necessidade de ambientes de simulação e avaliação experimental como elementos estratégicos no processo de concepção das novas aplicações IoT massivas, por possibilitarem que as soluções possam ser testadas em uma escala estendida.

A proposta ora apresentada neste artigo busca suprir duas importantes demandas do cenário de IoT massivo, que podem ser traduzidas nos seguintes desafios de domínio ("*Domain Challenges*"):

DC_1 - implantação de infraestruturas de testes que permitam simular, de maneira simples e rápida, cenários diversos compostos por uma quantidade expressiva de fontes contextuais, possivelmente heterogêneas, gerando dispositivos adaptáveis a cenários particulares idealizados pelos desenvolvedores de soluções IoT. Tais infraestruturas devem viabilizar a provisão de uma extensa lista de dispositivo que representem o domínio de aplicação, de forma simplificada e com baixo custo.

DC_2 - interação individual e de forma homogênea com dispositivos IoT das mais diversas tecnologias, de modo a abstrair as especificidades de implementação de cada dispositivo em termos de seus componentes (sensores e atuadores), tecnologias de hardware e software básico embarcado. No caso de cidades inteligentes, por exemplo, onde a integração de dados é fundamental, pode haver a necessidade de interagir com inúmeros dispositivos IoT, com características de hardware e funcionalidades completamente diversas, a fim de atender aos propósitos das aplicações. Implantar, portanto, serviços e aplicações que explorem as potencialidades que se abrem a partir da composição de múltiplas e diferentes fontes de dados pode ser desafiador, haja vista a necessidade de compreender a especificidade de cada dispositivo.

Iniciativas como [1], [2], [3] tem se dedicado à proposição de estratégias de uniformização na forma de interagir e conhecer as capacidades dos dispositivos. Lançada recentemente, a proposta apresentada em [4] tem se destacado como uma abordagem promissora nessa direção, especificando um padrão na forma de expor as capacidades dos dispositivos IoT (denominado *Thing Description*), que considera o enriquecimento semântico nas descrições através da associação à conceitos de ontologias e o uso de padrões de comunicação para acessar as suas capacidades, criando, assim, uma abstração no que se refere às especificidades de cada dispositivo IoT.

Uma análise como estes dois desafios são tratados por algumas propostas de infraestruturas de simulação reportadas na literatura é apresentada na Seção 3. Observa-se que, em uma visão integrada destes desafios, as plataformas de simulação para IoT massiva devem se comprometer com a promoção da flexibilidade e da padronização, como forma de ampliar a sua percepção como um importante aliado no processo de viabilização de produtos IoT por meio da experimentação.

III. TRABALHOS RELACIONADOS

Em [13] é apresentado o simulador de rede de sensores sem fio TOSSIM, construído com o objetivo específico de simular dispositivos que suportam a execução do sistema operacional TinyOS [14]. O ambiente apresentado é capaz de simular o comportamento da rede e dos seus nós sensores, capturando aspectos temporais e de consumo de energia. Os autores afirmam que sua estrutura é capaz de executar milhares de nós em uma única máquina. Embora o trabalho proposto possibilite a granularidade no nível de dispositivo, o seu foco está na investigação de protocolos de comunicação que atendam aos requisitos do domínio de aplicação com a maior economia de energia possível, haja vista as fortes restrições de recursos de hardware dos dispositivos. Em relação aos desafios de domínio elencados na seção anterior, ambos os desafios DC_1 e DC_2 não são atendidos devido à limitação da provisão heterogênea de nós e ausência de descrição semântica.

Semelhante ao TOSSIM, a plataforma de simulação COOJA [15], desenvolvida para o sistema operacional Contiki [16], fornece um simulador completo para o software presente no nó sensor, onde é possível simular a composição de uma rede de sensores, incluindo aspectos como o link de rádio,

interferências de comunicação sem fio, entre outros. O COOJA possibilita a simulação de nós que estejam executando outras plataformas, como TinyOS [14], ampliando a heterogeneidade da rede simulada. Contudo, o COOJA, de maneira análoga ao TOSSIM, também se mostra focado em simular aspectos de protocolos de comunicação no contexto de redes de sensores sem fio. Em relação aos desafios do domínio, o DC_1 é parcialmente atendido, pois embora apresente suporte a outras plataformas os cenários simulados estão restritos ao contexto de redes de sensores sem fio. O desafio DC_2 não é contemplado nesse simulador.

Em [17] é proposto o MAMMOTH, um emulador de IoT em larga escala, que pode ser usado para criar cenários de experimentos, implantá-los em uma plataforma de teste e monitorar os resultados. Um dos propósitos é reproduzir os problemas de comunicação presentes em um ambiente real ao qual os dispositivos IoT estão conectados, simulando links de rádio para cada dispositivo. Os autores relataram que foi possível executar 10.000 nós em uma única máquina virtual. Contudo, embora esta proposta possibilite a geração massiva de nós, questões importantes relacionadas a heterogeneidade e características individuais de cada nó se mostram como desafios de pesquisa em aberto. Em relação aos desafios do domínio, o DC_1 é parcialmente atendido. Embora explicitamente possibilite a geração massiva de nós, a heterogeneidade é limitada à plataforma proprietária que emula os nós sensores. Em relação ao desafio DC_2 o trabalho o contempla parcialmente dada a tentativa de uniformizar a interação com o dispositivo por meio da definição de um protocolo de acesso padronizado, no entanto, o trabalho não apresenta estratégias de expor as funcionalidades do dispositivo também de maneira padronizada.

O trabalho descrito em [18] apresenta o MobIIoTSim - uma proposta de simulador IoT que é implementado como um aplicativo móvel para a plataforma Android e capaz de instanciar um ou mais dispositivos. Dados do sensor do dispositivo simulado são valores gerados aleatoriamente dentro do intervalo especificado pelo usuário. No entanto, o usuário deve cadastrar manualmente cada dispositivo no aplicativo móvel, informando os parâmetros requeridos, como nome, frequência, valor máximo e mínimo gerado pelo sensor simulado, sendo que cada dispositivo possui apenas um sensor simulado. Em relação à interação do simulador com as aplicações, o ambiente não proporciona uma forma de explicitar as características do dispositivo simulado e limita a forma de interação a um broker MQTT. Em relação aos desafios do domínio, o desafio DC_1 é parcialmente atendido, visto que não há suporte à geração massiva automatizada. No que se refere ao desafio DC_2 o trabalho o contempla parcialmente em relação à tentativa de uniformizar a interação com o dispositivo. Isto é feito por meio de um protocolo previamente definido (o MQTT) e do estabelecimento de uma forma comum de expor os dados gerados a partir dos dispositivos, no caso, usando o formato de dados JSON.

Em [19] é apresentada uma evolução do MobIIoTSim. No novo sistema é possível a criação de grupos de dispositivos,

permitindo que seja simulado uma quantidade extensa de dispositivos do mesmo tipo, ampliando as possibilidades de simulações devido ao aumento da escalabilidade neste cenário. Embora esta nova versão do simulador possibilite parametrizar a quantidade de instâncias, formando grupos de dispositivos simulados, a diversidade gerada ainda é limitada. Em relação aos desafios do domínio, DC_1 é parcialmente atendido, em razão da heterogeneidade dos dispositivos IoT ser limitada nesta abordagem. Da mesma maneira, o desafio DC_2 também é atendido parcialmente, pois a estratégia de interação com o dispositivo IoT é a mesma do trabalho anterior [18].

Em [20] é proposto um simulador, denominado iFogSim, para modelar ambientes de IoT e Fog com foco na medição do impacto de técnicas de gerenciamento de recursos, considerando latência, congestionamento de rede, consumo de energia e custo. Em [21] o iFogSim foi estendido para possibilitar a modelagem e simulação de cenários abordando diferentes políticas de *data placement*. Percebe-se em [20] e [21] que o principal foco está relacionado à simulação de uma infraestrutura fog e as diversas variáveis relacionadas. As características do dispositivo IoT, como a sua forma de interação e diversidade e descrição semântica, não foram explorados nesses trabalhos. O desafio de domínio DC_1 é parcialmente atendido, pois possibilita a simulação de sensores e atuadores, embora o foco desta proposta esteja em prover a infraestrutura de comunicação e lidar com os obstáculos deste contexto. Em relação à DC_2 o artigo não é claro quanto à maneira de interagir com cada dispositivo IoT simulado.

O simulador proposto em [22] também tem como foco a computação fog e objetiva avaliar o projeto e implantação de aplicações neste contexto. Embora apresente contribuições importantes, a concepção da infraestrutura demonstra ser o elemento central da arquitetura proposta. O dispositivo IoT é um elemento secundário da arquitetura, sendo apresentado apenas como um estudo de caso. Por fim, em relação aos desafios de domínio, DC_1 é parcialmente atendido, não respondendo à demanda de provisão de um cenário massivo e heterogêneos de dispositivos IoT. O desafio DC_2 não é contemplado neste trabalho.

A Tabela I resume o nível de suporte oferecido por cada um dos trabalhos a estes desafios de pesquisa. Os desafios marcados como “-” representam aqueles que não foram identificados ou para os quais não houve informações suficientes nos estudos. Células contendo o símbolo “+” representam desafios observados e cumpridos pelos trabalhos. Para os casos em que os desafios foram parcialmente observados, a célula contém o símbolo “*”.

Tabela I
DESAFIOS DO DOMÍNIO.

Desafios	[13]	[15]	[17]	[18]	[19]	[20]	[22]
DC_1	-	*	*	*	*	*	*
DC_2	-	-	*	*	*	-	-

IV. REQUISITOS DO SISTEMA

A análise de trabalhos relacionados permitiu estabelecer um panorama representativo de como os simuladores se estruturam em termos de organização funcional e, em particular, como eles se comprometem com os desafios de domínio do ambiente de IoT massiva. Isto permitiu refinar os desafios de domínio em termos de requisitos essenciais que devem ser atendidos pela solução de geração massiva de dispositivos IoT aqui proposta.

Requisito r_1 : Adaptabilidade. Como resultado da análise, vimos que os trabalhos [18], [19], [20], [22] atendem em parte ao DC_1 , oferecendo suporte limitado à adaptação da simulação no que se refere ao domínio de aplicação. Neste sentido, acreditamos que a proposta de solução deva possibilitar a personalização individual dos dispositivos IoT simulados em relação às suas características, a fim de atender plenamente às necessidades do domínio de aplicação. Como este é um recurso relevante para a maioria dos cenários de interesse da presente proposta, derivamos o nosso primeiro requisito r_1 , o qual se refere à possibilidade de definir como devem ser gerados os dispositivos IoT simulados, incluindo os sensores, atuadores e intervalo de geração de valores de cada elemento.

Requisito r_2 : Escalabilidade heterogênea. Os trabalhos [15], [18], [19], [22], [20] atendem parcialmente ao DC_1 , oferecendo suporte limitado em relação à geração automatizada de conjuntos heterogêneos de dispositivos IoT. A solução aqui proposta deve permitir simular cenários diversificados, compostos por dispositivos com plataformas de hardware e software diversas que respondam às demandas do domínio da aplicação. Como este é um recurso relevante para viabilizar a concepção de novos produtos e serviços, derivamos nosso segundo requisito r_2 , que diz respeito à capacidade de geração massiva de dispositivos IoT heterogêneos.

Requisito r_3 : Modelo de interação. Como visto, o desafio DC_2 foi uma preocupação presente em alguns dos trabalhos analisados, tendo sido parcialmente abordado por [17], [18], [19] no que se refere à maneira pela qual as aplicações interagem com os dispositivos IoT, abstraindo detalhes específicos de cada componente da infraestrutura. Desta preocupação, derivamos o nosso requisito r_3 , que trata da interface de interação com as capacidades expostas dos dispositivos.

Requisito r_4 : Descrição semântica de dispositivos IoT. Um requisito de fundamental importância no cenário de IoT massiva, não explorado pelos trabalhos correlatos analisados, se refere à existência de uma maneira uniforme de descrever, com fundamentação semântica, os dispositivos IoT em termos de seus componentes, funcionalidades oferecidas e protocolos suportados. Esta descrição abre caminho para resolver problemas de integração, configuração e gerenciamento de dispositivos, além de permitir interações de forma intuitiva. Derivamos, assim, o nosso requisito r_4 , que se refere à especificação semântica de dispositivos IoT.

A análise dos trabalhos correlatos mostra que existe uma carência de ambientes de simulação que atendam, de forma integrada, aos requisitos estabelecidos para um cenário de

Internet das Coisas massiva. A Tabela II resume o nível de suporte oferecido por cada um dos trabalhos a estes requisitos.

Tabela II
REQUISITOS.

Requisitos	[13]	[15]	[17]	[18]	[19]	[20]	[22]
r_1	-	-	-	*	*	*	*
r_2	-	*	*	*	*	*	*
r_3	-	-	*	*	*	-	-
r_4	-	-	-	-	-	-	-

V. ARQUITETURA

Esta seção apresenta o projeto da arquitetura desenvolvida. A seção se inicia com a apresentação do modelo de informação proposto, seguido da descrição dos componentes da arquitetura. A seção também aborda questões e escolhas tecnológicas realizadas durante o seu desenvolvimento.

A. Dispositivo IoT Virtual - Modelo de Informação

O modelo de informação proposto, apresentado na Figura 1, constitui o elemento básico sobre o qual todo o funcionamento do ambiente de simulação se fundamenta. O modelo proposto foi inspirado na recomendação W3C Web of Things [23].

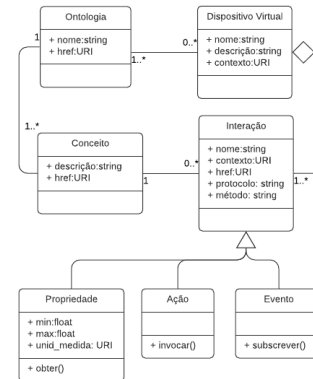


Figura 1. Modelo de informação.

Neste modelo, a abstração ou conceito de modelagem central é o *dispositivo virtual*. Um dispositivo virtual consiste na representação em software de um elemento do domínio IoT com as mesmas funcionalidades de um dispositivo físico real. Um dispositivo virtual é composto por *interações*, que podem ser entendidas como uma camada de abstração que fornece uma interface e protocolos padronizados a fim de acessar as funcionalidades providas pelo dispositivo IoT. As funcionalidades podem ser dos seguintes tipos: (i) *propriedade* - podem ser usadas para detectar e controlar parâmetros, como obter o valor atual ou definir um estado de uma operação; (ii) *ação* - modelam a invocação de processos físicos, mas também podem ser usadas para abstrair chamadas remotas; ou (iii) *evento* - usados para o modelo de comunicação por *push* em que notificações, eventos discretos ou fluxos de valores são enviados de forma assíncrona para o receptor.

Propriedades fornecem uma interface de abstração ao processo de interação com sensores físicos de um dispositivo. Por exemplo, a obtenção de dados de um determinado sensor requer, muitas vezes, a implementação de funções capazes de interpretar o sinal elétrico oriundo do sensor, obtido através da detecção de modificações de alguma variável de um ambiente, convertendo esses dados em um valor de uma escala conhecida. Neste sentido, essa abstração habilita às aplicações o acesso aos dados de interesse oriundos de um sensor presente em um dispositivo IoT.

Ações fornecem uma abstração quanto ao processo de interação com atuadores físicos de um dispositivo. Um atuador requer um conjunto de instruções específicas a fim de converter sinais elétricos fornecidos como entrada, em ações mecânicas. Por exemplo, um *buzzer* requer sinais elétricos que representem a frequência e o tempo que este atuador deve propagar ondas mecânicas.

Um evento representa um modelo *publish/subscribe* em que aplicações podem se inscrever a determinados eventos ou sequência de episódios de interesse. Por exemplo, a ocorrência de uma arritmia cardíaca detectada a partir dos dados coletados dos dispositivos.

No modelo de informação proposto, um dispositivo virtual deve estar sempre associado a *ontologias*, que constituem a abordagem semântica que dá o suporte necessário para a definição formal e precisa dos elementos que compõem o dispositivo virtual.

Por sua vez, uma *interação* é entendida como uma abstração que fornece uma interface e protocolos de comunicação padronizados para acessar as funcionalidades providas pelo dispositivo IoT, tais como seus sensores, atuadores e eventos. Cada interação também deve estar associada a um conceito de uma ontologia. Neste trabalho, ontologias são representadas por vocabulários OWL - Web Ontology Language [24].

B. Arquitetura proposta

A arquitetura proposta é ilustrada na Figura 2, sendo constituída de dois elementos principais: o *Virtual Thing Generator* e o *Virtual Thing*.

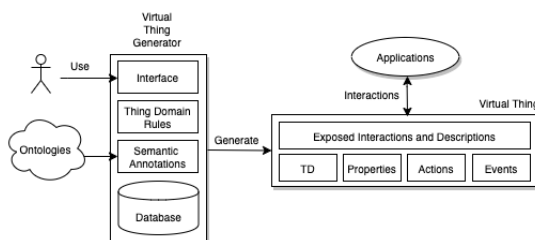


Figura 2. Visão geral da arquitetura proposta.

O *Virtual Thing Generator* é o elemento da arquitetura responsável por todo processo de geração de dispositivos virtuais. Por meio do componente *Interface* o usuário fornece os parâmetros requeridos pelo seu algoritmo de geração, tais como o local onde serão armazenados os arquivos gerados referentes aos dispositivos virtuais, a quantidade desejada de

dispositivos e a quantidade máxima de interações (propriedades, ações e eventos) que um dispositivo virtual deve possuir.

Conjuntos de *Interações* são previamente registradas no *Database* do *Virtual Thing Generator*, sendo possível adaptá-los conforme os cenários idealizados, o que atende ao Requisito r_1 . Por exemplo, o usuário é capaz de definir quais são as características internas dos *Virtual Things* a serem gerados pelo ambiente, como o universo de sensores, atuadores e eventos que podem compor o dispositivo gerado.

Thing domain rules diz respeito às regras do domínio referente ao propósito do dispositivo. A concepção de um dispositivo é comumente o reflexo dos requisitos elencados a fim de atender às necessidades de um domínio. Nesse sentido, especificidades do domínio influenciam as decisões de implementação das funcionalidades do dispositivo. Uma típica interação revelada pelas características do domínio é o evento.

O componente *Semantic Annotation*, tem como foco o enriquecimento semântico da descrição de um dispositivo IoT. Para tal, ele faz uso de uma linguagem padrão de descrição de dispositivos, que uniformiza a sintaxe e possibilita a definição semântica dos diversos aspectos de interesse do dispositivo. Na arquitetura proposta, a semântica é concretizada por meio da associação de conceitos de ontologias aos elementos que compõe a Coisa (*Thing*), tornando a definição de tais elementos mais precisa. Essa associação é um processo realizado pelo desenvolvedor de IoT, que efetua uma análise sobre o conceito mais adequado a ser vinculado. Nesta etapa, é preconizado o reuso de ontologias existentes, tais como SAREF [25], M3-Lite [26] e OM [27].

O mecanismo de geração de dispositivos virtuais procura atender ao Requisito r_2 e é feito maneira que segue. Inicialmente, a quantidade de interações de um dispositivo é especificada, sendo esta um valor inteiro, aleatório, entre 1 e a quantidade máxima de interações definida pelo usuário. Em seguida, cada interação é selecionada por meio de um sorteio entre as interações previamente registradas e, então, é associada ao novo dispositivo. Cada dispositivo consiste em um arquivo javascript gerado pela aplicação, baseado na implementação do projeto *Eclipse Thingweb*¹. Cada dispositivo virtual gerado possui um comportamento equivalente a um dispositivo real no que se refere à composição e exposição do *Thing Description* e a disponibilização das interações por meio dos protocolos previstos na especificação W3C Web of Things.

O *Virtual Thing* é o produto final do ambiente proposto, sendo gerado a partir dos parâmetros definidos pelo usuário no componente *Virtual Thing Generator*. Ele consiste na representação virtual de um dispositivo IoT, sendo composto tipicamente por elementos computacionais que simulam e capacidade de processamento, comunicação, sensoriamento e/ou elementos atuadores presentes em um dispositivo IoT.

O *Virtual Thing* é exposto para as aplicações por meio do componente *Exposed Interaction and Description*, que

¹<https://projects.eclipse.org/proposals/eclipse-thingweb>

busca revelar às aplicações interessadas a descrição completa de uma Coisa (*Thing*). Esta descrição (*Thing Description - TD*) é realizada através de uma linguagem de descrição de dispositivos (DDL), explicitando os conceitos associados e a forma pela qual é possível interagir com as suas propriedades, ações e eventos. Estas informações favorecem o processo de seleção de dispositivos de IoT e promovem um elevado nível de abstração, pois as aplicações precisam apenas conhecer a interface de diálogo, isto é, a forma de interagir e os protocolos de comunicação suportados. Este elemento da arquitetura propicia o atendimento aos Requisitos r_3 e r_4 .

A arquitetura proposta provê o acesso às interações e descrições disponibilizadas pelo dispositivo IoT por meio dos protocolos HTTP, CoAP e MQTT. A adoção de cada protocolo é uma decisão particular de cada desenvolvedor IoT, conforme os requisitos elencados e o propósito do dispositivo IoT.

VI. AVALIAÇÃO

Como forma de validar o ambiente proposto, foi realizada uma geração de dispositivos virtuais, que foram hospedados em uma instância EC2 da AWS (t2.micro), como pode ser visto na Figura 3. Nesta figura, cada linha representa um dispositivo IoT virtual gerado.

```

▼ 0: "http://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:8080/device_101a4202"
▼ 1: "http://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:8080/device_102a3096"
▼ 2: "http://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:8080/device_102a1759"

```

Figura 3. Dispositivos gerados pelo ambiente de simulação IoT Massivo.

Além da implementação da lógica necessária para prover o funcionamento de seus componentes (ações, propriedades e eventos), tornando-os acessíveis por meio de um particular protocolo, o algoritmo presente em cada dispositivo virtual gerado também possui a capacidade de compor e expor o TD, como preconizado pela arquitetura conceitual. A exposição do *Thing Description* ocorre por meio do protocolo HTTP; assim, quando o usuário quiser conhecer todas as características do dispositivo, basta realizar uma requisição utilizando o método GET do protocolo HTTP no endereço exposto do dispositivo.

A Figura 4 apresenta o trecho inicial do *Thing Description* de um dispositivo gerado pela aplicação, onde é possível observar informações como título, descrição, contexto e tipo. O contexto é parte fundamental do processo de descrição semântica do dispositivo e seus componentes, sendo o local onde são referenciadas as ontologias que são utilizadas para distinguir conceitualmente as características e capacidades de um dispositivo IoT Virtual.

Para cada dispositivo a ser gerado, o simulador realiza uma escolha aleatória das interações disponíveis no banco de dados, considerando a quantidade máxima de interações definidas pelo utilizador do sistema. Como resultado, são gerados dispositivos IoT virtuais com extensa composição de interações, fornecendo um ambiente com ampla diversidade de

```

title: "device_101a4202"
description: "Description test"
▼ @context:
  0: "https://www.w3.org/2019/wot/td/v1"
  1:
    saref: "https://w3id.org/saref"
    om: "http://www.wurvoc.org/vocabularies/om-1.8/"
    m3-lite: "http://purl.org/iot/vocab/m3-lite"
    geo: "http://www.w3.org/2003/01/geo/wgs84_pos"

```

Figura 4. Contexto de um Dispositivo gerado.

dispositivos. A Figura 5 ilustra as propriedades disponíveis em um determinado dispositivo gerado. Neste exemplo, há quatro sensores disponíveis - oxigênio, dióxido de carbono, som e luminosidade. O algoritmo presente no dispositivo gera, para cada propriedade, um valor aleatório entre um valor mínimo e máximo que é definido no momento do registro da mesma, sendo os valores gerados acessíveis por meio dos protocolos HTTP e CoAP.

```

▼ properties:
  ▶ oxygen: (-)
  ▶ carbon_dioxide: (-)
  ▶ sound: (-)
  ▶ light: (-)

```

Figura 5. Propriedades do dispositivo virtual gerado.

A Figura 6 apresenta a propriedade oxigênio em mais detalhes, onde é possível observar a associação com conceitos de ontologias existentes com objetivo de descrever semanticamente a propriedade. Neste exemplo, o tipo é associado com o conceito Sensor da ontologia SAREF [25] e a unidade de medida retornada pela propriedade é concentração em partes por milhão, definida pelo conceito PPM da ontologia m3-lite [26]. O valor gerado pela propriedade pode ser obtido por meio dos protocolos HTTP e CoAP e o conteúdo retornado é em formato JSON. Essa abordagem propicia a compreensão precisa e a maneira pelo qual será possível interagir com as funcionalidades presentes no dispositivo.

```

▼ properties:
  ▼ oxygen:
    type: "saref:Sensor"
    description: "null"
    unit of measurement: "m3-lite:PPM"
    observable: false
    readOnly: true
    writeOnly: false
    ▼ forms:
      ▼ 0:
        href: "http://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:8080/device_114a2276/properties/oxygen"
        contentType: "application/json"
        ▼ op:
          0: "readproperty"
          htv:methodName: "GET"
      ▼ 1:
        href: "coap://ec2-3-18-220-42.us-east-2.compute.amazonaws.com:5683/device_114a2276/properties/oxygen"
        contentType: "application/json"
        ▼ op:
          0: "readproperty"

```

Figura 6. Detalhes da propriedade oxigênio do dispositivo virtual gerado.

O diagrama de sequência que demonstra este cenário está representado na Figura 7. Uma requisição de valores do sensor de oxigênio é usada para demonstrar o acesso às propriedades de um dispositivo IoT virtual no estudo de caso. O usuário inicia o processo, utilizando método GET do protocolo CoAP para fazer a requisição ao dispositivo conectado (1); o dispositivo conectado à rede recebe a solicitação e executa a lógica necessária para a obtenção dos valores de oxigênio fornecidos pelo algoritmo (2), e retorna um valor aleatório entre o mínimo e máximo previamente estabelecido no registro desta interação (3). Ao receber as leituras, o dispositivo processa os valores recebidos em formato JSON e os envia ao usuário que requisitou usando o protocolo CoAP (4).

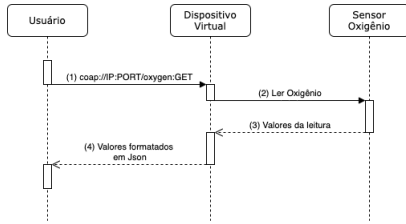


Figura 7. Diagrama de sequência da interação com a propriedade oxygen do dispositivo virtual gerado.

Como visto, atuadores são elementos importantes no ambiente de IoT. A Figura 8 apresenta um exemplo de representação das propriedades de um *buzzer*. A ontologia SAREF [25] foi utilizada para caracterizar semanticamente o tipo (Actuator) e a unidade de medida retornada (On/Off). O valor do estado atual do *buzzer* pode ser obtido por meio dos protocolos HTTP e CoAP, acessado via URL. Os valores devem estar em formato JSON. A mudança de estado do atuador pode ser realizada por meio da URL exposta utilizando o método POST dos protocolos HTTP e CoAP.

```

▼ properties:
  ▼ buzzer:
    type: "saref:Actuator"
    description: "null"
    unit of measurement: "saref:OnOffFunction"

```

Figura 8. Detalhes do atuador buzzer do dispositivo virtual gerado.

Com intuito de avaliar o desempenho na geração de dispositivos virtuais, foi realizado um experimento em que foi medido do tempo do processo de geração de tais dispositivos (T_{total}), considerando a quantidade de dispositivos, o tempo para persistência em banco de dados (T_{bd}) e o tempo para gerar os arquivos (T_{arq}) referentes aos dispositivos virtuais. O resultado desta avaliação pode ser visto na Tabela 1.

$$T_{total} = T_{bd} + T_{arq} \quad (1)$$

O objetivo deste experimento foi quantificar o tempo de geração de dispositivos. Observa-se que o tempo total tende a um comportamento linear, demonstrando viabilidade da geração massiva de dispositivos IoT.

Outro experimento realizado foi a implementação de uma aplicação que faz uso de um *Virtual Thing* - VT gerado pela arquitetura proposta. O VT foi hospedado em uma plataforma de

Tabela III
TEMPO PARA GERAÇÃO DE DISPOSITIVOS VIRTUAIS.

Dispositivos	T_{bd}	T_{arq}
10	00:00:01.456	00:00:00.071
50	00:00:04.087	00:00:00.238
100	00:00:06.811	00:00:00.574
500	00:00:29.419	00:00:02.285
1000	00:00:56.512	00:00:04.588
5000	00:05:02.169	00:00:33.955

nuvem computacional e o experimento consistiu na requisição de informações de uma determinada propriedade (no caso, sensor de oxigênio), conforme representado na Figura 7.

Esta interação do tipo propriedade foi previamente registrada no banco de dados para fornecer um intervalo de valores entre zero e 200 PPM (Partículas por Milhão) m3-lite:PPM. Foram executadas cinco etapas, nas quais a aplicação realizou 100, 500, 1000, 5000 e 10000 requisições com intervalo de 10ms entre cada requisição, demonstrando um exemplo uso intensivo deste dispositivo IoT virtual. Todas as requisições estão representadas na Figura 9. Esta figura apresenta um gráfico de dispersão que ilustra todas as requisições realizadas, indicando o tempo de resposta e o valor de sensoriamento obtido.

Como pode ser observado, o dispositivo IoT gerado pela arquitetura proposta demonstrou ser capaz de suportar um uso intensivo por aplicações, com a maior parte tempo de resposta entre 300 ms e 1200 ms, tendo um conjunto minoritário de requisições com tempo de resposta superior a 4800ms.

VII. CONSIDERAÇÕES FINAIS

Este trabalho apresentou um ambiente de simulação para Internet das Coisas Massiva. A característica principal deste ambiente é a adoção da recomendação W3C Web of Things como estratégia para uniformizar a maneira de expor as funcionalidades de um dispositivo e a forma de interagir com as funcionalidades disponíveis no mesmo. O objetivo é fornecer uma infraestrutura simples, porém útil, dotada de funcionalidades importantes para apoio às pesquisas que exploram cenários com um conjunto extenso de dispositivos IoT, com diversidade de interações (propriedades, ações e eventos).

Os testes realizados evidenciaram o potencial do ambiente na provisão massiva de dispositivos virtuais, e permitiram comprovar a sua flexibilidade na definição dos elementos que compõem uma Coisa (*Thing*) e a eficiência na maneira de expor as características de dispositivos IoT. Desta forma, foi possível explorar funcionalidades de dispositivos em um contexto próximo de um cenário real.

Direções futuras de pesquisa incluem a implementação da geração de dispositivos IoT virtuais com suporte à interação do tipo evento e o desenvolvimento de um modelo que simule o consumo de energia, CPU, memória RAM e rede, possibilitando obter um índice relacionado às perdas no processo de transmissão.

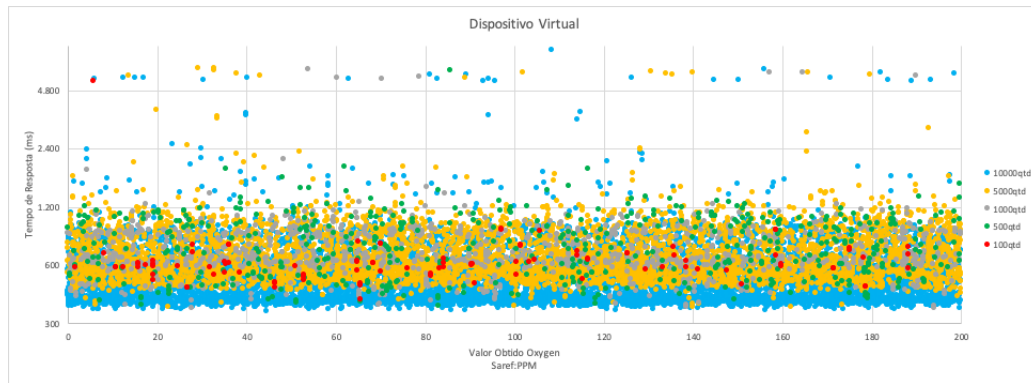


Figura 9. Teste de requisições da propriedade oxygen de um dispositivo virtual

AGRADECIMENTOS

Os autores são gratos à Fundação de Amparo à Pesquisa e Inovação do Espírito Santo - FAPES (Universal - No. 21/2018) pelo financiamento deste projeto de pesquisa.

REFERÊNCIAS

- [1] A. E. Khaled, A. Helal, W. Lindquist, and C. Lee, "IoT-DDL-Device Description Language for the 'T' in IoT," *IEEE Access*, vol. 6, pp. 24 048–24 063, 2018.
- [2] C. Chen and A. Helal, "Device integration in soda using the device description language," in *2009 Ninth Annual International Symposium on Applications and the Internet*. IEEE, 2009, pp. 100–106.
- [3] S. K. Datta and C. Bonnet, "Describing things in the Internet of Things: From CoRE link format to semantic based descriptions," *2016 IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW 2016*, no. i, pp. 1–2, 2016.
- [4] K. T. M. M. C. V. K. M. Kaebisch S. (2020, feb) Web of Things (WoT) Thing Description. [Online]. Available: <https://www.w3.org/TR/wot-thing-description/>
- [5] S. C. de Souza and J. G. Pereira Filho, "Semantic interoperability in iot: A systematic mapping," in *Internet of Things, Smart Spaces, and Next Generation Networks and Systems*. Springer, 2019, pp. 53–64.
- [6] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "IoT-Lite: A Lightweight Semantic Model for the Internet of Things," in *2016 Intl IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCCom/IoP/SmartWorld)*, 2016, pp. 90–97.
- [7] A. Gyrard, A. Zimmermann, and A. Sheth, "Building IoT based applications for Smart Cities: How can ontology catalogs help?" *IEEE Internet of Things Journal*, 2018. [Online]. Available: <http://dx.doi.org/10.1109/JIOT.2018.2854278>
- [8] K.-H. Le, S. K. Datta, C. Bonnet, and F. Hamon, "WoT-AD: A Descriptive Language for Group of Things in Massive IoT," *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pp. 257–262, 2019.
- [9] G. Z. Papadopoulos, J. Beaudaux, A. Gallais, T. Noel, and G. Schreiner, "Adding value to wsn simulation using the iot-lab experimental platform," in *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2013, pp. 485–490.
- [10] G. Z. Papadopoulos, A. Gallais, G. Schreiner, E. Jou, and T. Noel, "Thorough iot testbed characterization: From proof-of-concept to repeatable experimentations," *Computer Networks*, vol. 119, pp. 86–101, 2017.
- [11] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis *et al.*, "Smart-santander: Iot experimentation over a smart city testbed," *Computer Networks*, vol. 61, pp. 217–238, 2014.
- [12] M. Presser, L. Vestergaard, and S. Ganea, "Smart city use cases and requirements," *CityPulse Project Deliverable D*, vol. 2, p. 1, 2014.
- [13] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: Accurate and scalable simulation of entire tinyos applications," in *Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 126–137.
- [14] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer *et al.*, "Tinyos: An operating system for sensor networks," in *Ambient intelligence*. Springer, 2005, pp. 115–148.
- [15] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*. IEEE, 2006, pp. 641–648.
- [16] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki-a lightweight and flexible operating system for tiny networked sensors," in *29th annual IEEE international conference on local computer networks*. IEEE, 2004, pp. 455–462.
- [17] V. Looga, Z. Ou, Y. Deng, and A. Ylä-Jääski, "Mammoth: A massive-scale emulation platform for internet of things," in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, vol. 3. IEEE, 2012, pp. 1235–1239.
- [18] T. Pflanzner, A. Kertész, B. Spinnewyn, and S. Latré, "Mobiotsim: Towards a mobile iot device simulator," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*. IEEE, 2016, pp. 21–27.
- [19] A. Kertesz, T. Pflanzner, and T. Gyimothy, "A mobile iot device simulator for iot-fog-cloud systems," *Journal of Grid Computing*, vol. 17, no. 3, pp. 529–551, 2019.
- [20] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.
- [21] M. I. Naas, J. Boukhobza, P. R. Parvedy, and L. Lemarchand, "An extension to ifogsim to enable the design of data placement strategies," in *2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*. IEEE, 2018, pp. 1–8.
- [22] I. Lera, C. Guerrero, and C. Juiz, "Yafs: A simulator for iot scenarios in fog computing," *IEEE Access*, vol. 7, pp. 91 745–91 758, 2019.
- [23] K. T. M. M. C. V. K. M. Kaebisch S. (2020, feb) W3C Web of Things (WoT). [Online]. Available: <https://www.w3.org/WoT/>
- [24] D. L. McGuinness, F. Van Harmelen *et al.*, "Owl web ontology language overview," *W3C recommendation*, vol. 10, no. 10, p. 2004, 2004.
- [25] L. Daniele, F. den Hartog, and J. Roes, "Created in close interaction with the industry: the smart appliances reference (saref) ontology," in *International Workshop Formal Ontologies Meet Industries*. Springer, 2015, pp. 100–112.
- [26] R. Agarwal, D. G. Fernandez, T. Elsaleh, A. Gyrard, J. Lanza, L. Sanchez, N. Georgantas, and V. Issarny, "Unified iot ontology to enable interoperability and federation of testbeds," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE, 2016, pp. 70–75.
- [27] H. Rijgersberg, M. van Assem, and J. Top, "Ontology of units of measure and related concepts," *Semantic Web*, vol. 4, no. 1, pp. 3–13, 2013.