UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES

GUSTAVO TSUYOSHI ARIGA

WotPy: Solução de Problemas e Exemplo de Uso

São Paulo
Julho de 2023

UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES

GUSTAVO TSUYOSHI ARIGA

WotPy: Solução de Problemas e Exemplo de Uso

Relatório de Atividades apresentada à Escola de Artes, Ciências e Humanidades, da Universidade de São Paulo, como parte dos requisitos exigidos na disciplina ACH 2017 – Projeto Supervisionado ou de Graduação I, para obtenção do título de Bacharel em Sistemas de Informação.

Área de concentração: Metodologia e Técnicas da Computação

Orientador: Prof. Dr. Fábio Nakano

Coorientador: Prof. Dr. José de Jesús Pérez-Alcázar

Modalidade: TCC Longo (1 ano) - individual

São Paulo

Julho de 2023

Glossário

API: *Application Programming Interface* - conjunto de regras e protocolos que permite que diferentes softwares se comuniquem entre si. 2

Arquitetura: estrutura geral e organização de um sistema ou aplicação, incluindo seus componentes, padrões e princípios subjacentes. 2

Descoberta WoT: *WoT Discovery* - mecanismo de descoberta de recursos e serviços na arquitetura do WoT. 2

Descrição de Coisas: *Thing Descriptions* - descrições de metadados e interfaces de rede de coisas (dispositivos) no contexto do W3C-WoT. 2

Diretrizes de Segurança e Privacidade: *Security and Privacy Guidelines* - orientações e melhores práticas para a implementação segura de dispositivos e serviços no contexto do WoT. 2

Endpoint SPARQL: interface de acesso a um grafo de conhecimento que permite consultas e recuperação de informações usando a linguagem de consulta SPARQL. 2

Gateway: dispositivo ou *software* que conecta redes ou sistemas diferentes, permitindo a comunicação e a transferência de dados entre eles. 2

Grafo de conhecimento: estruturas de dados que representam conhecimento e informações em forma de grafos, onde os nós representam entidades e as arestas representam as relações entre elas. 2

Handlers: componentes de *software* responsáveis por lidar com solicitações, eventos ou tarefas específicas em um sistema ou aplicação. 2

Interconexão: conexão e integração de diferentes sistemas, dispositivos ou redes para permitir a comunicação e a troca de informações entre eles. 2

Interoperabilidade: capacidade de diferentes sistemas ou dispositivos se comunicarem, trocarem informações e trabalharem juntos de forma eficiente e eficaz. 2

IoT: Internet das Coisas (*Internet of Things*) - interconexão de dispositivos físicos (coisas) que são capazes de coletar e trocar dados por meio de uma rede. 2

Microcontrolador: dispositivo eletrônico que incorpora um microprocessador, memória e periféricos em um único chip. 2

Modelos de Vinculação: *Binding Templates* - modelos que fornecem orientações para definir interfaces de rede para protocolos específicos e ecossistemas de IoT no contexto do W3C-WoT. 2

Protocolo de comunicação: conjuntos de regras e formatos de dados que governam a troca de informações entre sistemas ou dispositivos de rede. 2

Scripting API: API baseada em JavaScript que permite a implementação da lógica de aplicação das coisas no contexto do W3C-WoT. 2

Servidor: computador ou sistema que fornece serviços, recursos ou funcionalidades a outros dispositivos ou sistemas, conhecidos como clientes, por meio de uma rede. 2

W3C: *World Wide Web Consortium* - consórcio internacional que desenvolve padrões e diretrizes para a *World Wide Web*, visando promover sua acessibilidade, usabilidade e interoperabilidade. 2

Web: *World Wide Web* - sistema de informação e documentos interconectados, acessíveis por meio da Internet e navegadores da Web. 2

WoT: Web das Coisas (*Web of Things*) - extensão da Web para abranger a interconexão e interação de dispositivos físicos por meio de padrões e tecnologias da Web. 2

WoTPy: Web das Coisas Python (*Web of Things Python*) - *gateway* experimental baseado no W3C-WoT. 2

Resumo

ARIGA, Gustavo Tsuyoshi. **WotPy: Solução de Problemas e Exemplo de Uso**. Julho de 2023. 36 p. Relatório de Atividades – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2023.

O propósito deste estudo foi auxiliar no progresso do WoTPy, um *gateway* experimental fundado no W3C-WoT, com a intenção de aprimorar a comunicação entre aparelhos IoT (Internet das Coisas) de diversas marcas. Para atingir tal propósito, metas específicas foram estipuladas, como o exame das normas do W3C-WoT, a escolha e investigação das bibliotecas e utensílios ligados às dependências do WoTPy, o enfrentamento de desafios de instalação, a execução de testes e a confirmação da validade das colaborações, além da elaboração de exemplos práticos e documentação detalhada. A compreensão das especificações possibilitou sua implementação correta e eficiente, enquanto a seleção e domínio das bibliotecas e ferramentas adequadas garantiram suporte aos protocolos de comunicação e facilidade de integração. A resolução dos problemas de instalação tornou o WoTPy facilmente implantável e configurável em diferentes ambientes e sistemas operacionais, favorecendo sua adoção por desenvolvedores e usuários finais. Os exemplos de uso e a documentação detalhada desenvolvidos auxiliaram outros desenvolvedores e usuários na implementação do WoTPy em seus projetos IoT. Conclui-se, portanto, que o WoTPy é um *gateway* funcional e eficaz, promovendo a interoperabilidade e integração de dispositivos IoT. Para futuros avanços, a junção do WoTPy com grafos de conhecimento será avaliada, o que permite a apresentação das habilidades dos sensores e suas observações através de um *endpoint SPARQL*, expandindo as opções de conexão e compartilhamento de dados entre aparelhos IoT. Esta pesquisa auxiliou na evolução do campo da Internet das Coisas, apresentando uma alternativa pragmática e eficaz para aprimorar a comunicação e unificação de dispositivos IoT.

Palavras-chaves: Web das Coisas (WoT). World Wide Web Consortium (W3C). WoTPy.

Abstract

ARIGA, Gustavo Tsuyoshi. **WotPy: Troubleshooting and Usage Example.** Julho de 2023. 36 p. Activity Report – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2023.

The purpose of this study was to contribute to the development of WoTPy, an experimental gateway based on W3C-WoT, with the aim of improving interoperability among IoT (Internet of Things) devices from different manufacturers. To achieve this goal, specific objectives were established, including the analysis of W3C-WoT specifications, the selection and study of libraries and tools related to WoTPy dependencies, addressing installation challenges, conducting tests and validating contributions, as well as developing practical examples and detailed documentation. Understanding the specifications enabled the correct and efficient implementation of WoTPy, while the selection and mastery of appropriate libraries and tools ensured support for communication protocols and ease of integration. Resolving installation problems made WoTPy easily deployable and configurable in various environments and operating systems, facilitating its adoption by developers and end users. The developed usage examples and detailed documentation assisted other developers and users in implementing WoTPy in their IoT projects. In conclusion, WoTPy is a functional and effective gateway, promoting interoperability and integration of IoT devices. For future advancements, the integration of WoTPy with knowledge graphs will be explored, enabling the presentation of sensor capabilities and observations through a SPARQL endpoint, expanding the options for connectivity and data sharing among IoT devices. This research has contributed to the advancement of the Internet of Things field, providing a pragmatic and effective solution to enhance communication and unification of IoT devices.

Keywords: Web das Coisas (WoT). World Wide Web Consortium (W3C). WoTPy.

Lista de figuras

Sumário

1 Introdução

Neste capítulo inicial, a contextualização, o tema da pesquisa, a tese, os propósitos e a abordagem de investigação do projeto serão expostos. Além disso, uma descrição sucinta da disposição do documento será proporcionada.

1.1 Contextualização

Este projeto de estudo está vinculado à área de Internet das Coisas (IoT), correspondendo à conexão de equipamentos inteligentes em uma rede mundial. A IoT detém a capacidade de revolucionar vários ramos, como saúde, indústria, agricultura e cidades inteligentes, gerando vantagens como automação, economia de energia e monitoramento à distância.

Contudo, a comunicação entre dispositivos IoT de diferentes marcas ainda representa um obstáculo considerável. A ausência de padronização e a variedade de protocolos e interfaces impedem a interação e integração entre estes equipamentos. Isso restringe a habilidade de elaborar soluções extensas e escaláveis que aproveitem todo o potencial da IoT.

Diante deste panorama, o projeto de estudo propõe enfrentar o desafio da interoperabilidade, concentrando-se na execução e melhoria do WoTPy, um *gateway* experimental baseado no W3C-WoT. A meta é desenvolver uma resposta que facilite a interação e a união entre dispositivos IoT de diferentes marcas, seguindo os padrões e orientações estipulados pelo W3C.

1.2 Tema da Pesquisa

A fundamentação para a execução deste projeto de pesquisa reside na exigência de vencer as barreiras da interoperabilidade na IoT. A ausência de padronização e a variedade de protocolos e interfaces impedem a interação e a integração entre os dispositivos IoT, limitando o potencial da tecnologia.

O questionamento da pesquisa reside em criar uma solução que promova a interoperabilidade entre dispositivos IoT de variadas marcas usando o WoTPy. A meta é

permitir que estes dispositivos interajam e colaborem de forma transparente, possibilitando a construção de respostas abrangentes e escaláveis na IoT fazendo o uso do WoTPy.

1.3 Tese

Baseado na análise do tema da pesquisa, a suposição apresentada é que a execução e aprimoramento do WoTPy como um *gateway* experimental fundado no W3C-WoT possa simplificar a interação e a união entre dispositivos IoT de variadas marcas. Crê-se que essa resposta, alinhada aos padrões e orientações do W3C, pode colaborar para a superação dos obstáculos da interoperabilidade na IoT.

1.4 Propósitos

O propósito central deste projeto de pesquisa é desenvolver e aperfeiçoar o WoTPy como um *gateway* experimental fundado no W3C-WoT, visando aprimorar a interoperabilidade entre dispositivos IoT de variadas marcas. Para atingir este propósito central, os seguintes propósitos específicos foram estipulados:

- Entender as normas do W3C-WoT e suas principais partes, como Descrição das Coisas (*Thing Descriptions*), Modelos de Vinculação (*Binding Templates*) e *Scripting API*.
- Selecionar e investigar as bibliotecas e ferramentas necessárias para a execução do WoTPy.
- Solucionar desafios de instalação e configuração do WoTPy, visando simplificar sua implantação em diferentes ambientes.
- Realizar testes e validações para garantir a funcionalidade e qualidade do WoTPy.
- Criar um exemplo de uso práticos do WoTPy e documentação para auxiliar desenvolvedores e usuários na implementação e utilização do *gateway*.

1.5 Abordagem de Pesquisa

Este projeto de pesquisa adota um método de pesquisa aplicada, combinando elementos de pesquisa exploratória e desenvolvimento de *software*. A metodologia incluiu as seguintes etapas:

1. Revisão bibliográfica sobre o W3C-WoT, IoT e interoperabilidade.
2. Análise das normas do W3C-WoT e estudo aprofundado de suas principais partes.
3. Seleção e estudo das bibliotecas e ferramentas relacionadas ao WoTPy.
4. Desenvolvimento e aperfeiçoamento do WoTPy, incluindo solução de desafios de instalação, testes e validações.
5. Elaboração de exemplos de uso práticos do WoTPy e criação de documentação detalhada.

1.6 Disposição do Documento

Este documento está disposto da seguinte forma:

- Capítulo 1: Introdução - expõe a contextualização, a fundamentação, a suposição, os propósitos e a abordagem de pesquisa do projeto.
- Capítulo 2: Objetivos - expõe os objetivos gerais e específicos do projeto.
- Capítulo 3: Revisão Bibliográfica - aborda os conceitos fundamentais relacionados ao W3C-WoT, IoT e interoperabilidade.
- Capítulo 4: Metodologia - descreve detalhadamente as etapas e os procedimentos adotados na execução do projeto.
- Capítulo 5: Resultados - expõe os resultados obtidos durante a execução e aprimoramento do WoTPy.
- Capítulo 6: Discussão - analisa e discute os resultados obtidos, destacando suas contribuições e limitações.
- Capítulo 7: Conclusão - sintetiza os principais pontos discutidos no trabalho, expõe as conclusões alcançadas e sugere possíveis direções futuras.

2 Objetivos

2.1 Objetivo Geral

O propósito principal deste projeto foi auxiliar no progresso do WoTPy Mangas *et al.* (2022), um *gateway* experimental sustentado no W3C-WoT. Este *gateway* visa incrementar a interoperabilidade entre equipamentos IoT de distintos fabricantes, promovendo a comunicação e a associação destes em uma única plataforma ou sistema.

2.2 Objetivos Específicos

Para alcançar o objetivo principal, os seguintes objetivos específicos foram definidos:

1. Avaliar as especificações do W3C-WoT Lagally *et al.* (2023) e entender os conceitos chaves e requisitos para a implementação do WoTPy;
2. Escolher e examinar as bibliotecas e instrumentos necessários para as dependências do WoTPy;
3. Solucionar dificuldades de instalação, assegurando que o WoTPy possa ser implementado e configurado sem problemas em diferentes contextos e sistemas;
4. Realizar testes e validações das melhorias realizadas;
5. Criar exemplos de uso e documentação minuciosa para auxiliar programadores e usuários na implementação do WoTPy em seus projetos de IoT;

2.3 Desdobramento

Com a realização destes objetivos no primeiro semestre de 2023, espera-se tratar, no segundo semestre do mesmo ano, a integração do WoTPy com grafos de conhecimento. Este aprimoramento visa intensificar a interoperabilidade dos equipamentos IoT, permitindo a divulgação das habilidades dos sensores e suas observações por meio de um *endpoint SPARQL*.

3 Revisão Bibliográfica

Neste capítulo, será realizado uma análise de literatura referente à Web das Coisas (WoT) e sua estrutura sugerida pelo World Wide Web Consortium (W3C). Também será discutido os desafios envolvidos na interoperabilidade entre redes IoT e o papel dos *gateways* neste cenário.

3.1 A Web das Coisas e a Estrutura do W3C

A Web das Coisas (WoT) é um conceito que tem recebido atenção crescente recentemente. O W3C descreve a WoT como a conexão de sensores, atuadores, objetos físicos, locais e até pessoas por meio da Internet. O intuito da WoT é usar as tecnologias da Web para simplificar o desenvolvimento de aplicações e serviços que envolvam esses dispositivos e suas representações virtuais (GROUP, 2015b; GROUP, 2015a).

A estrutura proposta pelo W3C para a WoT determina os elementos que compõem esta rede interligada. Ela ressalta a relevância da infraestrutura e dos protocolos da Internet para possibilitar a comunicação entre os dispositivos. Ademais, a estrutura enfatiza a existência de dispositivos de borda, também conhecidos como *gateways*, que exercem um papel crucial na interoperabilidade entre as redes IoT e a Web (MATSUKURA *et al.*, 2023).

A estrutura da WoT define os dispositivos de borda como pontos de conexão entre a rede interna e a Internet. Estes dispositivos são encarregados de realizar a computação e o processamento de dados perto dos sensores e atuadores, frequentemente em ambientes com recursos limitados. Este método, conhecido como computação de borda ou *fog computing*, permite a implementação de soluções de interoperabilidade e segurança no nível do *gateway* (STIRBU, 2008; GYRARD *et al.*, 2017; García Mangas; Suárez Alonso, 2019).

3.2 O Framework WoTPy

Dentro do contexto da WoT, o WoTPy surge como um framework desenvolvido no meio acadêmico com a finalidade de implementar as recomendações do W3C. O WotPy é um conjunto de ferramentas que permite o desenvolvimento de *gateways* IoT, englobando

diferentes protocolos de comunicação, como HTTP, MQTT e CoAP. Ele foi concebido para facilitar a interoperabilidade entre dispositivos e oferecer suporte a funcionalidades cruciais, como privacidade, segurança, descoberta de dispositivos e interfaces de usuário (García Mangas; Suárez Alonso, 2019).

4 Metodologia

O processo metodológico empregado neste trabalho descreve as fases e estratégias adotadas. Ele foi concebido com a finalidade de alcançar os resultados propostos e responder às questões da pesquisa formuladas.

4.1 Estratégia de Pesquisa

A estratégia de pesquisa escolhida para este trabalho foi baseada em uma mescla de pesquisa exploratória, revisão bibliográfica e desenvolvimento prático. A pesquisa exploratória foi executada para obter um entendimento mais profundo do tema, identificar os conceitos-chave e tendências, e averiguar as soluções existentes. A revisão bibliográfica foi executada para examinar e analisar as principais publicações, artigos acadêmicos e normas associadas ao WoTPy e ao W3C-WoT. Isso forneceu fundamentação teórica para o trabalho e identificou brechas ou possibilidades de aprimoramento. O desenvolvimento prático incluiu a implementação e teste do WoTPy e a elaboração de exemplos de uso e documentação extensa.

4.2 Coleta de Informações

A coleta de informações foi feita por meio de várias fontes, como artigos acadêmicos, publicações, documentação oficial do W3C-WoT e WoTPy, e experimentos práticos. A revisão literária foi realizada para reunir dados relevantes sobre o W3C-WoT, as tecnologias envolvidas, os padrões e as práticas recomendadas. A documentação oficial do W3C-WoT Lagally *et al.* (2023) e do WoTPy Mangas *et al.* (2018) foi analisada para entender as especificações, APIs e diretrizes sugeridas.

4.3 Elaboração do WoTPy

A elaboração do WoTPy foi conduzida com base nas especificações e diretrizes do W3C-WoT. Primeiramente, foi feita uma análise minuciosa das especificações do W3C-WoT para compreender os conceitos essenciais e requisitos. Em seguida, foram escolhidas as

bibliotecas e ferramentas apropriadas como dependências do WoTPy. Foram solucionados problemas de instalação e configuração para assegurar a facilidade de implementação em diferentes ambientes e sistemas operacionais. A elaboração do WoTPy foi feita utilizando a linguagem de programação Python e seguindo as melhores práticas de desenvolvimento de *software*.

4.4 Verificação e Testes

A verificação do WoTPy foi feita por meio de um conjunto de testes disponíveis no repositório wot-py Mangas *et al.* (2022).

4.5 Elaboração do Exemplo de Uso e Documentação

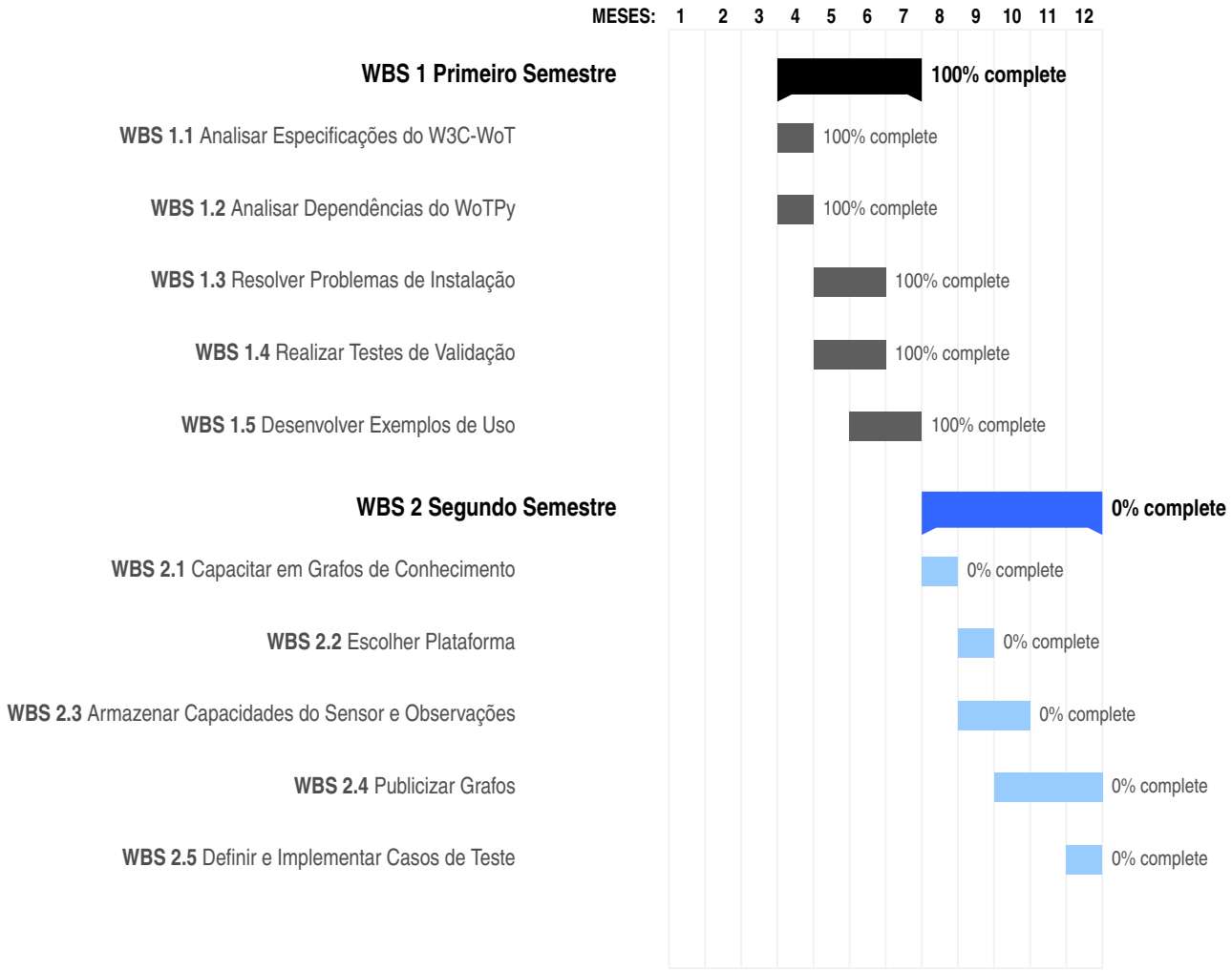
Para ajudar os desenvolvedores e usuários na implementação do WoTPy em seus projetos IoT, foi elaborado um exemplo de uso prático e documentação. O exemplo de uso abrange cenários comuns de aplicação do WoTPy, demonstrando como utilizar suas funcionalidades e integrar dispositivos IoT. A documentação oferece informações sobre a instalação e configuração, para a utilização do WoTPy.

4.6 Programação

O trabalho foi realizado em fases sequenciais, com uma programação estabelecida para guiar o progresso e a organização das atividades, conforme demonstrado na figura 1.

Cada fase foi planejada para ser finalizada dentro de um período de tempo específico, possibilitando o avanço constante do trabalho e o cumprimento dos prazos fixados.

Figura 1 – Cronograma



5 Resultados

Os resultados esperados, delineados no plano de atividades Ariga e Nakano (2023), foram atingidos de maneira adequada durante a execução do trabalho, para mais informações aceesar este diretório (ARIGA, 2023f). A seguir, são descritos os desfechos alcançados:

Entendimento aprofundado das especificações do W3C-WoT: Uma análise completa das especificações do W3C-WoT foi realizada, gerando um entendimento detalhado dos conceitos e requisitos principais para a execução do WoTPy. Isso habilitou a implementação correta e eficaz do *gateway*, seguindo as normas definidas pelo consórcio.

Escolha e domínio de bibliotecas e ferramentas adequadas: As bibliotecas e ferramentas mais apropriadas para o desenvolvimento do WoTPy foram escolhidas, considerando a compatibilidade com os protocolos de comunicação necessários e a facilidade de integração com outros projetos IoT. O domínio dessas ferramentas foi essencial para o êxito do trabalho.

Resolução dos problemas de instalação do WoTPy: Problemas relacionados à instalação do WoTPy foram identificados e solucionados, resultando em um processo simplificado e eficaz para implantar e configurar o *gateway* em diferentes ambientes e sistemas operacionais. Isso simplificou a adoção do WoTPy por parte dos desenvolvedores e usuários finais, assegurando uma experiência mais suave.

Confirmação das contribuições por meio de testes: Uma série completa de testes foi conduzida para confirmar as contribuições feitas ao WoTPy. Esses testes garantiram a qualidade e a funcionalidade do *gateway*, comprovando sua conformidade com os requisitos e sua capacidade de lidar com diferentes casos de uso.

Criação de exemplos de uso e documentação: Foi desenvolvido um exemplo prático de uso do WoTPy, demonstrando sua aplicabilidade em cenários de IoT. Além disso, uma documentação foi produzida, oferecendo orientações claras e recursos adicionais para facilitar a execução do WoTPy em projetos IoT. Isso contribuiu para a disseminação e adoção do *gateway* pela comunidade.

5.1 Especificações do W3C-WoT

A Web das Coisas (WoT) é um esforço do World Wide Web Consortium (W3C) para estabelecer uma estrutura para conectar dispositivos e serviços na Internet das Coisas (IoT). O objetivo do WoT é possibilitar a interoperabilidade entre esses dispositivos, facilitando a comunicação e a integração padronizadas.

O W3C criou várias especificações que abordam diferentes aspectos do WoT. Essas especificações são essenciais para a execução e adoção do WoT, oferecendo diretrizes para descrever as características das Coisas (*Things*), definir interfaces de rede, realizar a descoberta de Coisas, implementar a lógica de aplicação e assegurar a segurança e a privacidade dos sistemas WoT.

- Descrição de Coisas (*Thing Description*) Kaebisch *et al.* (2020): estabelece um formato de dados interpretável por máquinas para descrever os metadados e as interfaces de rede das Coisas. Ela fornece uma base sólida para a interoperabilidade entre as Coisas e a Web.
- Modelos de Vinculação (*Binding Templates*) Koster e Korkan (2020): dão orientações sobre como definir interfaces de rede em Coisas para protocolos específicos e ecossistemas de IoT. Esses modelos são úteis para assegurar a compatibilidade e a integração de diferentes sistemas.
- Descoberta WoT (*WoT Discovery*) Cimmino *et al.* (2020): estabelece um mecanismo para distribuição de metadados das Coisas. Ela permite a localização e acesso a informações detalhadas sobre as Coisas, facilitando a descoberta e integração de dispositivos na rede.
- Modelos de Vinculação (*Scripting API*) Kis *et al.* (2020): permitem a implementação da lógica de aplicação das Coisas utilizando uma API JavaScript comum. Isso simplifica o desenvolvimento de aplicativos IoT e promove a portabilidade entre fornecedores e dispositivos.
- Diretrizes de Segurança e Privacidade (*Security and Privacy Guidelines*) McCool e Reshetova (2019): oferecem orientações para a implementação segura das Coisas e discutem questões relacionadas à segurança e à privacidade nos sistemas WoT. Essas diretrizes são importantes para proteger os dispositivos e os dados sensíveis envolvidos nas redes WoT.

5.1.1 Descrição de Coisas

A Descrição de Coisas representa um formato de dados que pode ser interpretado por máquinas, projetado para esclarecer os metadados e as interfaces de rede das Coisas na Web das Coisas. Esta norma, formulada pelo W3C, constitui uma base firme para incentivar a interoperabilidade entre as Coisas e a Web.

A Descrição de Coisas possibilita a descrição das características, propriedades, funcionalidades e interfaces de uma Coisa de maneira estruturada e padronizada. Essa descrição abrange informações detalhadas sobre como interagir com a Coisa, acessar seus recursos e interpretar os dados que ela disponibiliza.

O uso da Descrição de Coisas facilita a compreensão das capacidades e requisitos de uma Coisa específica por diferentes dispositivos e sistemas. Isso permite a criação de aplicações e serviços que podem interagir com uma vasta gama de Coisas de maneira transparente, independentemente do fabricante ou da tecnologia subjacente.

5.1.2 Modelos de Vinculação

Os Modelos de Vinculação fornecem orientações que detalham como definir interfaces de rede em Coisas para protocolos específicos e ecossistemas de IoT. Estes modelos são essenciais para garantir a compatibilidade e integração entre diferentes sistemas.

Cada protocolo de comunicação usado na Web das Coisas possui suas próprias especificidades e requisitos. Os Modelos de Vinculação fornecem um conjunto de instruções e recomendações para mapear funcionalidades e recursos de uma Coisa em um formato compatível com um protocolo específico.

Ao utilizar os Modelos de Vinculação, é possível cirar interfaces de rede para as Coisas que estejam em conformidade com os requisitos e padrões dos protocolos específicos. Isso facilita a comunicação e interação entre as Coisas e os sistemas que usam esses protocolos, garantindo maior interoperabilidade e uma integração mais harmoniosa.

Além disso, os Modelos de Vinculação ajudam a promover a reutilização de código e a padronização na implementação de interfaces de rede. Ao seguir estas orientações, é possível garantir que as Coisas sejam facilmente integradas em ecossistemas IoT existentes, evitando a necessidade de desenvolver soluções customizadas para cada protocolo.

A utilização dos Modelos de Vinculação também contribui para a escalabilidade e a flexibilidade dos sistemas IoT. Como esses modelos oferecem uma estrutura padronizada para a definição de interfaces de rede, é mais fácil adicionar novas Coisas ao ecossistema e integrá-las com os sistemas existentes, o que facilita a expansão dos ecossistemas IoT e o desenvolvimento de soluções mais amplas e interconectadas.

5.1.3 Descoberta WoT

A Descoberta WoT é um mecanismo que estabelece a distribuição de metadados das Coisas na Web das Coisas. Esta norma, desenvolvida pelo W3C, possibilita a localização e o acesso a informações detalhadas sobre as Coisas, facilitando a descoberta e a integração de dispositivos na rede.

A Descoberta WoT é crucial na Web das Coisas, pois permite que os dispositivos IoT sejam descobertos e identificados de maneira eficaz. Isso é especialmente importante em ambientes onde há uma grande quantidade de Coisas interconectadas, tornando a descoberta manual desses dispositivos impraticável.

Ao utilizar o mecanismo de Descoberta WoT, é possível adquirir metadados sobre as Coisas, como suas funcionalidades, propriedades, interfaces de rede e outros atributos relevantes. Essas informações detalhadas permitem que os desenvolvedores e os sistemas IoT identifiquem as Coisas que são relevantes para suas necessidades específicas.

Além disso, a Descoberta WoT permite a integração de dispositivos de diferentes fabricantes e tecnologias. Através da distribuição de metadados padronizados, os sistemas IoT podem identificar e interagir com as Coisas de maneira consistente, independentemente de suas características individuais.

A Descoberta WoT também contribui para a escalabilidade e flexibilidade dos sistemas IoT. Com a capacidade de localizar e acessar informações detalhadas sobre as Coisas, é mais fácil adicionar novos dispositivos à rede e integrá-los aos sistemas existentes, facilitando a expansão dos ecossistemas de IoT e o desenvolvimento de soluções mais completas e interconectadas.

5.1.4 Scripting API

A *Scripting API* é uma especificação que simplifica a criação de lógicas de aplicação IoT usando uma API JavaScript universal. Este método facilita a criação de aplicações IoT e incentiva a compatibilidade entre dispositivos e fornecedores.

Com a *Scripting API*, a lógica de aplicação IoT pode ser programada em JavaScript, uma linguagem conhecida por sua versatilidade e facilidade de uso. Isso elimina a necessidade de dominar linguagens de programação específicas para cada dispositivo ou fornecedor, tornando o desenvolvimento mais eficaz e acessível.

A *Scripting API* disponibiliza um conjunto de funcionalidades e métodos que possibilitam a interação com os dispositivos IoT, acessar seus recursos, enviar comandos e receber dados. Isso simplifica a criação de lógicas de negócios e a integração com outros componentes do sistema.

5.1.5 Diretrizes de Segurança e Privacidade

As Diretrizes de Segurança e Privacidade, são um conjunto de princípios essenciais que fornecem orientações para a construção segura de dispositivos IoT, abordando questões de segurança e privacidade nos sistemas WoT. Tais normas são fundamentais para a proteção de dispositivos e dados sensíveis presentes nas redes WoT.

A segurança e a privacidade são de grande importância na Web das Coisas, pois os dispositivos IoT estão cada vez mais presentes no cotidiano, manipulando um grande volume de dados sensíveis. As Diretrizes de Segurança e Privacidade fornecem recomendações e boas práticas para garantir a integridade, confidencialidade e disponibilidade dos dispositivos e dos dados na rede WoT.

Essas normas abrangem diversos aspectos de segurança, como autenticação, autorização, criptografia, gestão de chaves, proteção contra ataques e privacidade dos dados. Elas guiam na identificação e implementação de medidas de segurança adequadas em seus dispositivos IoT, mitigando riscos de exposição a ameaças e violações de privacidade.

5.2 Requisitos da W3C-WoTPy

O arquivo "setup.py" Ariga (2023c) define os requisitos para executar a biblioteca WotPy, como especificado pelo Consórcio World Wide Web (LAGALLY *et al.*, 2023). Esses requisitos são divididos em duas categorias: as obrigatórias e as opcionais.

Os principais requisitos obrigatórios listados no arquivo "setup.py" incluem:

- tornado: uma biblioteca assíncrona usada para desenvolver aplicativos web em Python. É usada pela WotPy para criar servidores HTTP e WebSocket.
- jsonschema: uma biblioteca que valida esquemas JSON e os dados JSON correspondentes. A WotPy usa essa biblioteca para verificar a validade das descrições de coisa (Thing Descriptions) recebidas e geradas.
- six: uma biblioteca que permite a escrita de código Python compatível com a versão 2 e 3. A WotPy usa essa biblioteca para assegurar a compatibilidade entre ambas as versões do Python.
- rx: uma biblioteca de programação reativa que facilita a escrita de código que responde assincronamente a mudanças de estado. É utilizada pela WotPy para suportar a API WoT Scripting e as interações com propriedades e eventos observáveis.
- python-slugify: uma biblioteca que transforma strings em "slug", um formato de texto que usa apenas caracteres ASCII, números e traços. A WotPy usa essa biblioteca para criar identificadores únicos para as coisas.

Além desses requisitos obrigatórios, existem alguns requisitos adicionais que a WoTPy usa, caso estejam presentes no sistema:

- aiocoap: uma biblioteca Python para o protocolo de transferência de dados Constrained Application Protocol (CoAP), utilizada pela WotPy para suportar o protocolo CoAP.
- hbmqtt: uma biblioteca Python que implementa o protocolo Message Queue Telemetry Transport (MQTT), usada pela WotPy para suportar o protocolo MQTT.
- websockets: uma biblioteca Python que suporta a comunicação via WebSocket, usada pela WotPy para suportar o protocolo WebSocket.
- zeroconf: uma biblioteca Python para suportar o protocolo DNS Service Discovery (DNS-SD), utilizada pela WotPy para descobrir serviços e dispositivos na rede.

Esses requisitos são verificados durante a execução e adicionados aos requisitos da biblioteca WoTPy, se estiverem presentes no sistema.

5.3 Resolução de Problemas e Verificação

Este relatório discute os problemas encontrados durante a execução e montagem do projeto WoTPy, disponível no GitHub (MANGAS *et al.*, 2022). Inicialmente, foi feito um fork do projeto, e o repositório Mangas *et al.* (2023) foi clonado para o computador.

5.3.1 Montagem do Projeto com Docker

Ao tentar montar o projeto usando "docker build .", encontrou-se um erro relacionado à versão do "pacote numpy". A mensagem de erro está descrita neste arquivo (ARIGA, 2023d).

A primeira tentativa de solução foi alterar a versão do Python no arquivo "Dockerfile" (ARIGA, 2023h).

Posteriormente, decidiu-se manter a versão original do Python (3.7) e, no arquivo "examples/benchmark/requirements.txt", foi revertida a versão modificada pelo "dependabot[bot]" (DEPENDABOT, 2022; ARIGA, 2023i).

A solução final foi remover a inclusão das dependências do diretório "/examples/benchmark" no "Dockerfile", mantendo a modificação anterior (ARIGA, 2023j). Esta decisão foi tomada considerando que o exemplo do "benchmark" não seria utilizado no projeto, e, portanto, as dependências relacionadas a ele não seriam necessárias.

5.3.2 Realização dos Testes

A realização dos testes do WoTPy foi iniciada com o comando "./pytest-docker-all.sh". Após a montagem, foi identificado o erro descrito neste arquivo Ariga (2023e). Para resolver esse problema, o arquivo setup.py foi modificado (ARIGA, 2023h).

Assim, o WoTPy pôde ser montado corretamente usando Docker e passou nos testes propostos no "pytest-docker-all.sh".

5.4 Elaboração do Exemplo Prático

O exemplo proposto demonstra a transmissão de dados do sensor ultravioleta (UV) de um microcontrolador ESP32 para um servidor Web das Coisas por meio do protocolo HTTP. O projeto usa a biblioteca WoTPy para construir o servidor e as interações com o WoT e o microcontrolador ESP32 emparelhado com um sensor UV ML8511.

O projeto é composto por dois arquivos de código principais: "server.py" e "main.py". O arquivo "server.py" (ARIGA, 2023b) configura o servidor WoT, revela uma Coisa que representa o sensor UV e define controladores personalizados para a leitura e gravação de dados do sensor UV. O arquivo "main.py" (ARIGA, 2023a), que é executado no ESP32, lê periodicamente os dados do sensor UV e os envia para o servidor WoT usando solicitações HTTP.

Este projeto atua como um exemplo de integração de dispositivos IoT usando os princípios do WoT, promovendo a comunicação e a interoperabilidade entre dispositivos e aplicações em um ecossistema de IoT. Para mais informações, acessar a documetação (ARIGA, 2023g).

5.4.1 Protocolo de Comunicação

O cenário inclui um dispositivo ESP32 equipado com um sensor ML8511 (sensor UV) atuando como cliente e um computador atuando como servidor com o uso da biblioteca WoTPy.

A comunicação entre o cliente ESP32 e o servidor WoT é estabelecida via protocolo HTTP, sendo uma opção adequada para esse contexto. A escolha do protocolo HTTP se justifica por diversos motivos que tornam essa abordagem viável e eficiente para o projeto em discussão.

A primeira justificativa para a escolha do protocolo HTTP é a familiaridade do autor com essa tecnologia, facilitando o desenvolvimento e evitando a necessidade de aprender um novo protocolo.

Além disso, o HTTP é amplamente utilizado na web e tem suporte de implementação para várias plataformas e linguagens de programação, incluindo o MicroPython usado no

ESP32. Existem bibliotecas e ferramentas disponíveis que facilitam a implementação da comunicação HTTP, permitindo um desenvolvimento mais eficaz.

O HTTP também suporta uma variedade de métodos de solicitação, como GET, POST, PUT e DELETE, permitindo que o cliente ESP32 faça solicitações para ler, gravar e excluir dados no servidor WoT. Essa flexibilidade é crucial para a interação entre o cliente e o servidor, permitindo a troca de informações necessárias para o funcionamento adequado do sistema WoT.

5.4.2 Cliente Web das Coisas

O arquivo "main.py" implementa um cliente Web of Things (WoT) no dispositivo ESP32, que interage com os sensores e envia os dados para um servidor WoT. O cliente WoT é responsável por coletar dados dos sensores e controlar os dispositivos conectados de maneira padronizada.

O cliente WoT utiliza o protocolo HTTP para se comunicar com o servidor WoT. No código, a biblioteca "*urequests*" é utilizada para realizar solicitações HTTP PUT ao servidor. Essas solicitações enviam os dados dos sensores no formato JSON para o servidor WoT, permitindo que os dados sejam armazenados e acessados pelos usuários.

No código, os sensores utilizados no ESP32, como o sensor UV, e suas configurações são definidos. Cada sensor tem uma identificação única e é associado a uma descrição no formato de Descrição das Coisas. Ela contém informações sobre o sensor, incluindo os links para acessar os dados no servidor WoT.

A função "send_sensor_data" é responsável por enviar os dados dos sensores para o servidor WoT. Esta função recebe como parâmetros a identificação do sensor, o tipo de sensor, o URL de acesso aos dados na Descrição das Coisas e os dados a serem enviados. A função realiza uma solicitação HTTP PUT ao URL especificado, enviando os dados em formato JSON. Se a solicitação for bem-sucedida, uma mensagem de sucesso é exibida. Caso contrário, uma mensagem de erro é exibida junto com o código de status da resposta.

No loop principal "main()", os valores dos sensores são lidos periodicamente. A cada iteração do loop, os dados dos sensores são obtidos e enviados para o servidor WoT usando a função "send_sensor_data". Esse processo permite que os dados dos sensores

sejam atualizados em tempo real no servidor WoT, possibilitando que os usuários acessem e usem essas informações de forma padronizada, conforme representado na figura 2.

5.4.3 Servidor Web das Coisas

O servidor Web of Things (WoT) é um elemento crucial na arquitetura WoT, pois expõe os dispositivos conectados e suas funcionalidades para serem descobertos e interagidos pelos clientes. O servidor WoT atua como uma ponte entre os dispositivos da Internet das Coisas e os aplicativos e serviços que desejam acessá-los.

O arquivo "server.py" implementa um servidor WoT que expõe um dispositivo responsável por fornecer os valores de um sensor UV. O servidor WoT permite a descoberta e interação com a Coisa através de solicitações HTTP padronizadas.

O servidor WoT usa a biblioteca Tornado e o protocolo HTTP para receber solicitações dos clientes e responder de acordo com as interações definidas na descrição da Coisa. A Coisa é definida por uma descrição no formato de um documento JSON, que especifica suas propriedades e comportamentos.

No código, é criado um servidor HTTP na porta especificada (HTTP_PORT) e um *Servient*, que é uma instância responsável por gerenciar os recursos WoT. A descrição da Coisa é definida, contendo o identificador (ID_THING) e as propriedades, como o sensor UV.

Para cada propriedade da Coisa, como o sensor UV, são definidos os manipuladores de leitura ("read_uv") e escrita ("write_uv"). O manipulador de leitura retorna o valor atual do sensor quando solicitado pelo cliente. O manipulador de escrita atualiza o valor do sensor quando recebe uma solicitação de escrita do cliente.

Ao iniciar o servidor, é criado o objeto WoT, e a Coisa é produzida com base na descrição. Em seguida, os manipuladores de leitura e escrita são associados à propriedade correspondente na Coisa. Isso permite que o servidor WoT responda às solicitações de leitura e escrita para o sensor UV.

Finalmente, a Coisa é exposta e o servidor inicia seu loop de eventos, aguardando as solicitações dos clientes e respondendo de acordo com as interações definidas, conforme representado na figura 2.

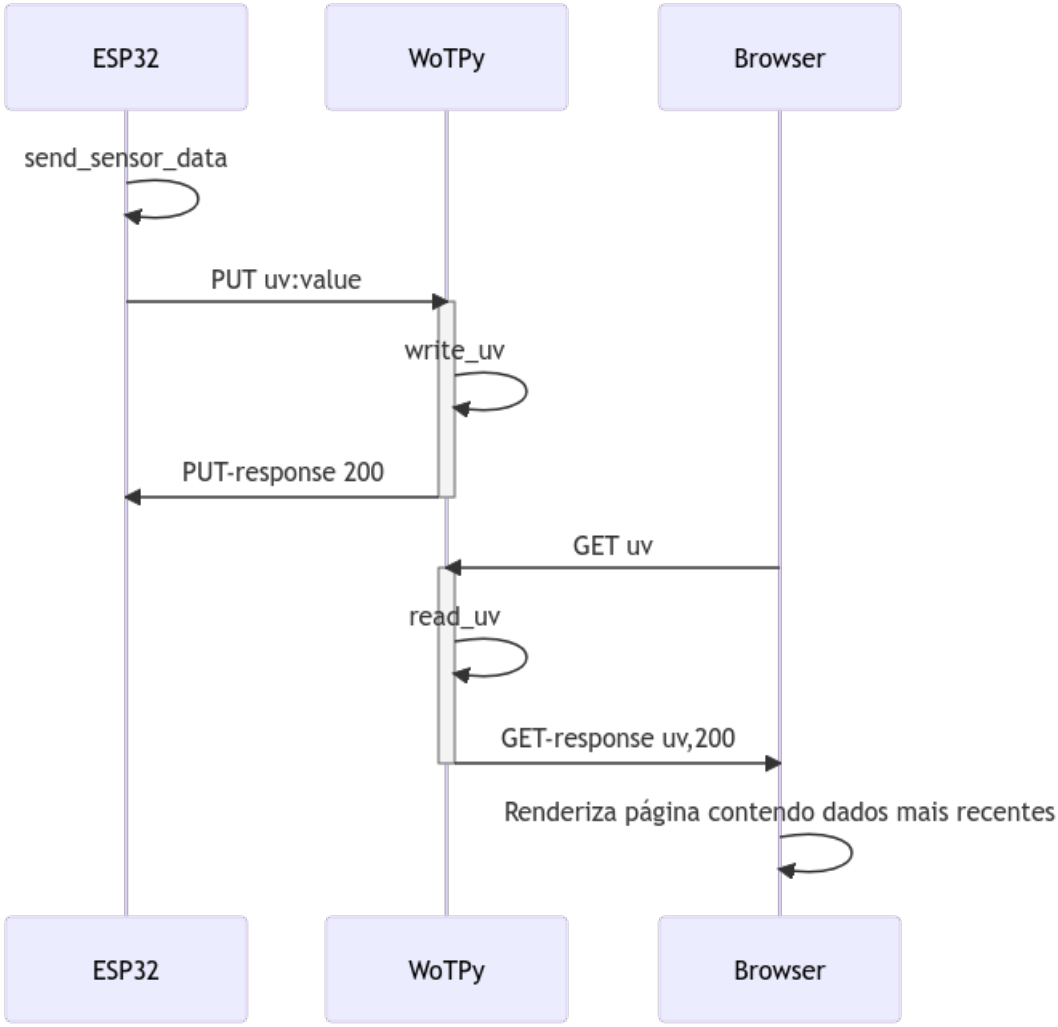


Figura 2 – O dispositivo (ESP32) executa um loop que, a cada 5 segundos executa a função send_sensor_data. Esta função envia uma requisição PUT para o servidor (WoTPy). Em resposta a essa requisição o servidor executa o handler (função) write_uv que contém o envio da resposta. Um usuário pode acessar a informação no servidor através do Browser. Entrar a URL na barra de endereços do Browser o faz enviar ao servidor uma requisição GET. Em resposta à requisição o servidor executa o handler (função) read_uv, que contém o envio da resposta - no caso, o valor de uv mais recente. O Browser recebe essa informação e renderiza a página contendo a informação.

6 Discussão

A seção de discussão tem como objetivo analisar e interpretar os resultados obtidos no trabalho, relacionando-os com a literatura existente e abordando os principais pontos e contribuições do estudo.

6.1 Análise dos Resultados

Os resultados obtidos demonstraram que o WoTPy é uma solução viável para a implementação de um *gateway* IoT baseado no W3C-WoT. Através da seleção cuidadosa das bibliotecas e ferramentas adequadas, foi possível desenvolver um gateway que suporta os principais protocolos de comunicação, como HTTP. Isso permite a integração de dispositivos IoT de diferentes fabricantes e facilita a interoperabilidade entre eles.

A compreensão das especificações do W3C-WoT foi essencial para o desenvolvimento do WoTPy. O estudo detalhado das especificações, como as Descrição das Coisas, Binding Templates e a *Scripting API*, permitiu a implementação correta e eficiente do *gateway*, garantindo a conformidade com os padrões estabelecidos.

A resolução dos problemas de instalação do WoTPy também se mostrou crucial. A simplificação do processo de instalação e configuração contribuiu para a fácil adoção do *gateway* em diferentes ambientes e sistemas operacionais. Além disso, os testes e validação realizados confirmaram a qualidade e a funcionalidade do WoTPy, fornecendo confiabilidade e confiança na sua utilização.

6.2 Comparação com Trabalhos Relacionados

Ao comparar o WoTPy com outros *gateways* IoT existentes, podemos observar suas vantagens e contribuições específicas. O WoTPy destaca-se por sua conformidade com as especificações do W3C-WoT, o que garante a interoperabilidade e a padronização no contexto da Web das Coisas. Além disso, sua flexibilidade e suporte aos principais protocolos de comunicação o tornam uma opção atrativa para a integração de dispositivos IoT heterogêneos.

A documentação detalhada e os exemplos de uso desenvolvidos também são diferenciais importantes do WoTPy. Esses recursos facilitam a compreensão e a utilização do gateway por parte dos desenvolvedores e usuários, contribuindo para a disseminação e adoção do projeto.

6.3 Limitações e Possíveis Melhorias

Durante o desenvolvimento deste trabalho, uma limitação identificada foi a incapacidade de utilizar o WoTPy no MicroPython. O MicroPython é uma implementação leve e eficiente do Python projetada para rodar em dispositivos com recursos limitados, como microcontroladores. No entanto, devido a diferenças de recursos e suporte a bibliotecas, o WoTPy não é atualmente compatível com o MicroPython.

Uma possível melhoria para contornar essa limitação seria a adaptação do WoTPy para ser compatível com o MicroPython. Isso permitiria que o gateway fosse implantado em dispositivos com recursos restritos, ampliando seu alcance e possibilitando sua utilização em uma variedade maior de cenários IoT. Essa adaptação envolveria ajustes nas dependências e otimizações específicas para o ambiente do MicroPython.

Além disso, uma outra melhoria potencial seria explorar alternativas de implementação específicas para o MicroPython. Isso poderia envolver a criação de uma versão simplificada do WoTPy ou o desenvolvimento de um gateway específico para dispositivos MicroPython. Essas abordagens permitiriam um melhor aproveitamento dos recursos limitados do MicroPython, garantindo a compatibilidade e a eficiência do gateway em tais dispositivos.

Essas melhorias seriam importantes para ampliar o alcance do WoTPy e torná-lo mais acessível a um maior número de dispositivos IoT, incluindo aqueles baseados no MicroPython. Ao superar a limitação atual e fornecer suporte para o MicroPython, o WoTPy se tornaria uma solução mais abrangente e versátil para a integração de dispositivos IoT em diferentes plataformas e ambientes.

6.4 Contribuições e Impacto

O presente trabalho contribui para o avanço da interoperabilidade e integração de dispositivos IoT no contexto da Web das Coisas. O desenvolvimento e a implementação do WoTPy como um gateway baseado no W3C-WoT oferecem uma solução prática e padronizada para a comunicação entre dispositivos de diferentes fabricantes.

As contribuições deste trabalho são relevantes tanto para a academia quanto para a indústria. Os resultados obtidos podem servir como base para futuras pesquisas e desenvolvimentos na área de IoT. Além disso, o WoTPy pode ser utilizado por empresas e profissionais que buscam criar soluções IoT interoperáveis e eficientes.

6.5 Considerações Finais

Através da metodologia adotada e da análise dos resultados, foi possível constatar que o WoTPy é uma solução eficaz para a implementação de um gateway IoT baseado no W3C-WoT. Suas funcionalidades, conformidade com as especificações do W3C e facilidade de uso o tornam uma opção viável e promissora no contexto da Web das Coisas.

O trabalho realizado contribui para a disseminação e adoção do WoTPy, bem como para a melhoria da interoperabilidade e integração de dispositivos IoT. Espera-se que as limitações identificadas possam ser superadas e que as melhorias sugeridas possam ser implementadas em trabalhos futuros.

7 Conclusão

Neste trabalho, foi abordada a solução de problemas e a criação de exemplos de uso da biblioteca WoTPy, um gateway experimental baseado no W3C-WoT. O objetivo principal foi contribuir para o desenvolvimento do WoTPy, visando melhorar a interoperabilidade e facilitar a comunicação e integração de dispositivos IoT de diferentes fabricantes.

Para atingir esse objetivo, foram realizadas diversas etapas ao longo do trabalho. Inicialmente, foram analisadas as especificações do W3C-WoT, compreendendo os principais conceitos e requisitos para a implementação do WoTPy. Em seguida, foram selecionadas e estudadas as bibliotecas e ferramentas necessárias para o desenvolvimento do projeto.

Durante o processo, foram identificados e resolvidos problemas de instalação do WoTPy, garantindo sua facilidade de implantação e configuração em diferentes ambientes e sistemas operacionais. Além disso, foram realizados testes e validações para garantir a qualidade e funcionalidade das contribuições feitas.

Um exemplo de uso prático do WoTPy foi desenvolvido, acompanhado por uma documentação detalhada, com o intuito de auxiliar desenvolvedores e usuários na implementação do WoTPy em seus projetos IoT. Essa documentação busca disseminar o conhecimento e facilitar a adoção do WoTPy pela comunidade.

Ao finalizar este trabalho, podemos afirmar que os resultados alcançados foram significativos. O WoTPy se tornou um gateway funcional e eficaz, capaz de melhorar a interoperabilidade entre dispositivos IoT de diferentes fabricantes. As contribuições realizadas, como a resolução de problemas de instalação, os testes e validações, e o desenvolvimento de exemplos de uso e documentação, agregaram valor ao projeto e facilitaram sua adoção e utilização.

Com base nos resultados obtidos e nas experiências adquiridas durante a realização deste trabalho, uma das áreas de desenvolvimento futuro é a integração do WoTPy com grafos de conhecimento. Essa integração visa aproveitar a estrutura semântica dos grafos para divulgar as capacidades dos sensores e suas observações por meio de um endpoint SPARQL. Dessa forma, seria possível realizar consultas semânticas para descobrir e explorar os recursos disponíveis nos dispositivos IoT, facilitando ainda mais a interoperabilidade dos dados coletados.

No geral, este trabalho contribuiu para avanços na área de Web das Coisas, promovendo a interoperabilidade e facilitando a integração de dispositivos IoT. Espera-se que o WoTPy e as contribuições realizadas possam ser adotados e utilizados por desenvolvedores e pesquisadores, impulsionando ainda mais o progresso nessa área em constante evolução.

Referências

ARIGA, G. T. *Arquivo main.py*. 2023. Disponível em: ⟨https://github.com/T16K/wot-py/blob/develop/examples/uv_sensor/main.py⟩. Citado na página 26.

ARIGA, G. T. *Arquivo server.py*. 2023. Disponível em: ⟨https://github.com/T16K/wot-py/blob/develop/examples/uv_sensor/server.py⟩. Citado na página 26.

ARIGA, G. T. *Arquivo setup.py*. 2023. Disponível em: ⟨https://github.com/T16K/wot-py/blob/develop/setup.py⟩. Citado na página 24.

ARIGA, G. T. *Construção do Projeto com Docker*. 2023. Disponível em: ⟨https://github.com/T16K/ACH2017/blob/a2630b977381a6c7aa3a71dd52ea72092d2a623b/Problema.md#constru%C3%A7%C3%A3o-do-projeto-com-docker⟩. Citado na página 25.

ARIGA, G. T. *Execução dos Testes*. 2023. Disponível em: ⟨https://github.com/T16K/ACH2017/blob/a2630b977381a6c7aa3a71dd52ea72092d2a623b/Problema.md#execu%C3%A7%C3%A3o-dos-testes⟩. Citado na página 25.

ARIGA, G. T. *READMES*. 2023. Disponível em: ⟨https://github.com/T16K/ACH2017/tree/main/READMES⟩. Citado na página 19.

ARIGA, G. T. *UV Sensor Data Transmission with ESP32 and WoT*. 2023. Disponível em: ⟨https://github.com/T16K/wot-py/blob/develop/examples/uv_sensor/README.md⟩. Citado na página 26.

ARIGA, G. T. *WoTPy Versão 1.1*. 2023. Disponível em: ⟨https://github.com/agmangas/wot-py/commit/b8e9d090e738b343a825cd404f63dde950954a70⟩. Citado na página 25.

ARIGA, G. T. *WoTPy Versão 1.2*. 2023. Disponível em: ⟨https://github.com/agmangas/wot-py/commit/61bec7348c6342c05fd8d2c3ccb21dad60aed58b⟩. Citado na página 25.

ARIGA, G. T. *WoTPy Versão 1.5*. 2023. Disponível em: ⟨https://github.com/agmangas/wot-py/commit/3dd4f7e428bd565970adf96f8f4482cf986496ea⟩. Citado na página 25.

ARIGA, G. T.; NAKANO, F. *Plano de Atividades*. 2023. Disponível em: ⟨https://github.com/T16K/ACH2017/blob/main/Plano%20de%20Atividades/main.pdf⟩. Citado na página 19.

CIMMINO, A.; MCCOOL, M.; TAVAKOLIZADEH, F.; TOUMURA, K. *Web of Things (WoT) Discovery*. 2020. Disponível em: ⟨https://www.w3.org/TR/wot-discovery/⟩. Citado na página 20.

DEPENDABOT. *Bump numpy from 1.15.4 to 1.22.0 in /examples/benchmark*. 2022. Disponível em: ⟨https://github.com/agmangas/wot-py/commit/ab14570927ccb1fcda5e7ffc415fda3c1ef2d00d⟩. Citado na página 25.

García Mangas, A.; Suárez Alonso, F. J. Wotpy: A framework for web of things applications. *Computer Communications*, v. 147, p. 235–251, 2019. ISSN 0140-3664. Disponível em: ⟨https://www.sciencedirect.com/science/article/pii/S0140366419304633⟩. Citado 2 vezes nas páginas 14 e 15.

GROUP, W. W. of T. C. *Main Page*. 2015. Disponível em: ⟨https://www.w3.org/community/wot/wiki/Main_Page#What_is_the_Web_of_Things.3F⟩. Acesso em: 29 de março de 2023. Citado na página 14.

GROUP, W. W. of T. I. *Terminology*. 2015. Disponível em: ⟨https://www.w3.org/WoT/IG/wiki/Terminology⟩. Acesso em: 29 de março de 2023. Citado na página 14.

GYRARD, A.; PATEL, P.; DATTA, S. K.; ALI, M. I. Semantic web meets internet of things and web of things: [2nd edition]. In: *Proceedings of the 26th International Conference on World Wide Web Companion*. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017. (WWW '17 Companion), p. 917–920. ISBN 9781450349147. Disponível em: ⟨https://doi.org/10.1145/3041021.3051100⟩. Citado na página 14.

KAEBISCH, S.; KAMIYA, T.; MCCOOL, M.; CHARPENAY, V. *Web of Things (WoT) Thing Description 1.1*. 2020. Disponível em: ⟨https://www.w3.org/TR/wot-thing-description11/⟩. Citado na página 20.

KIS, Z.; PEINTNER, D.; AGUZZI, C.; HUND, J.; NIMURA, K. *Web of Things (WoT) Scripting API*. 2020. Disponível em: ⟨https://www.w3.org/TR/wot-scripting-api⟩. Acesso em: 06 de abril de 2023. Citado na página 20.

KOSTER, M.; KORKAN, E. *Web of Things (WoT) Binding Templates*. 2020. Disponível em: ⟨https://www.w3.org/TR/wot-binding-templates⟩. Acesso em: 06 de abril de 2023. Citado na página 20.

LAGALLY, M.; MATSUKURA, R.; MCCOOL, M.; TOUMURA, K. *Web of Things (WoT) Architecture 1.1*. 2023. Disponível em: ⟨https://www.w3.org/TR/wot-architecture⟩. Acesso em: 06 de abril de 2023. Citado 3 vezes nas páginas 13, 16 e 24.

MANGAS, A. G.; CRESPO, A.; FATADEL; ROMANN, J.; RASHEED, M.; DANIELIBASETA; CI, T. *WoTPy*. 2018. Disponível em: ⟨https://agmangas.github.io/wot-py/index.html⟩. Citado na página 16.

MANGAS, A. G.; CRESPO, A.; FATADEL; ROMANN, J.; RASHEED, M.; DANIELIBASETA; CI, T. *WoTPy*. 2022. Disponível em: ⟨https://github.com/agmangas/wot-py⟩. Citado 3 vezes nas páginas 13, 17 e 25.

MANGAS, A. G.; CRESPO, A.; FATADEL; ROMANN, J.; RASHEED, M.; DANIELIBASETA; CI, T.; ARIGA, G. T. *WoTPy*. 2023. Disponível em: ⟨https://github.com/T16K/wot-py⟩. Citado na página 25.

MATSUKURA, R.; MCCOOL, M.; LAGALLY, M.; TOUMURA, K. *Web of Things (WoT) Architecture 1.1*. [S.l.], 2023. Https://www.w3.org/TR/2023/CR-wot-architecture11-20230119/. Citado na página 14.

MCCOOL, M.; RESHETOVA, E. *Web of Things (WoT) Security and Privacy Guidelines*. 2019. Disponível em: ⟨https://www.w3.org/TR/wot-security/⟩. Citado na página 20.

STIRBU, V. Towards a restful plug and play experience in the web of things. In: *2008 IEEE International Conference on Semantic Computing*. [S.l.: s.n.], 2008. p. 512–517. Citado na página 14.