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

arquitetura A arquitetura refere-se à estrutura geral e organização de um sistema ou aplicação, incluindo seus componentes, padrões e princípios subjacentes.. 2

Descrição de Coisas (Thing Descriptions) Thing Descriptions são descrições de metadados e interfaces de rede de coisas (dispositivos) no contexto do W3C-WoT. Elas fornecem informações sobre as capacidades, propriedades e serviços oferecidos pelas coisas.. 2

endpoint SPARQL Um endpoint SPARQL é uma interface de acesso a um grafo de conhecimento que permite consultas e recuperação de informações usando a linguagem de consulta SPARQL.. 2

Gateway Um gateway é um dispositivo ou software que conecta redes ou sistemas diferentes, permitindo a comunicação e a transferência de dados entre eles.. 2

grafos de conhecimento Grafos de conhecimento são 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 Handlers são componentes de software responsáveis por lidar com solicitações, eventos ou tarefas específicas em um sistema ou aplicação. Eles são responsáveis por receber, processar e responder a requisições ou ações específicas.. 2

interconexão A interconexão se refere à conexão e integração de diferentes sistemas, dispositivos ou redes para permitir a comunicação e a troca de informações entre eles.. 2

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

IoT IoT (Internet of Things) refere-se à interconexão de dispositivos físicos (coisas) que são capazes de coletar e trocar dados por meio de uma rede.. 2

microcontrolador Um microcontrolador é um dispositivo eletrônico que incorpora um microprocessador, memória e periféricos em um único chip. Ele é projetado para controlar funções específicas em sistemas eletrônicos.. 2

Modelos de Vinculação (Binding Templates) Binding Templates são modelos que fornecem orientações para definir interfaces de rede para protocolos específicos e ecossistemas de IoT no contexto do W3C-WoT. Eles ajudam a garantir a compatibilidade e a integração entre diferentes sistemas.. 2

protocolos de comunicação Protocolos de comunicação são conjuntos de regras e formatos de dados que governam a troca de informações entre sistemas ou dispositivos de rede.. 2

Scripting API Scripting API é uma API (Application Programming Interface) baseada em JavaScript que permite a implementação da lógica de aplicação das coisas no contexto do W3C-WoT. Ela simplifica o desenvolvimento de aplicativos IoT e promove a portabilidade entre fornecedores e dispositivos.. 2

servidor Um servidor é um computador ou sistema que fornece serviços, recursos ou funcionalidades a outros dispositivos ou sistemas, conhecidos como clientes, por meio de uma rede.. 2

W3C O W3C (World Wide Web Consortium) é um consórcio internacional que desenvolve padrões e diretrizes para a World Wide Web, visando promover sua acessibilidade, usabilidade e interoperabilidade.. 2

web A Web (World Wide Web) é um sistema de informação e documentos interconectados, acessíveis por meio da Internet e navegadores da web.. 2

WoT WoT (Web of Things) é um termo que se refere à extensão da Web para abranger a interconexão e interação de dispositivos físicos por meio de padrões e tecnologias da Web.. 2

WoTPy WoTPy é um gateway experimental baseado no W3C-WoT (Web of Things). Ele visa melhorar a interoperabilidade e a integração de dispositivos IoT de diferentes fabricantes, facilitando a comunicação e a interconexão em uma única aplicação ou sistema.. 2

Resumo

ARIGA, Gustavo Tsuyoshi. **WotPy: Solução de Problemas e Exemplo de Uso**. Julho de 2023. 39 Relatório de Atividades – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2023.

Este trabalho teve como objetivo contribuir para o desenvolvimento do WoTPy, um *gateway* experimental baseado no W3C-WoT, visando melhorar a interoperabilidade entre dispositivos IoT de diferentes fabricantes. Para alcançar esse objetivo, foram estabelecidos objetivos específicos, incluindo a análise das especificações do W3C-WoT, a seleção e estudo das bibliotecas e ferramentas relacionadas às dependências do WoTPy, a resolução de problemas de instalação, a realização de testes e validação das contribuições, e o desenvolvimento de exemplos de uso e documentação detalhada. Utilizando métodos de pesquisa bibliográfica, análise documental e experimentação, os resultados obtidos demonstraram que o WoTPy é capaz de facilitar a comunicação e integração entre dispositivos IoT, seguindo as especificações do W3C-WoT. 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. Como direcionamento para desenvolvimentos futuros, sugere-se a integração do WoTPy com grafos de conhecimento, possibilitando a divulgação das capacidades dos sensores e suas observações por meio de um *endpoint SPARQL*, ampliando as possibilidades de integração e troca de informações entre dispositivos IoT. Este trabalho contribuiu para o avanço da área de Internet das Coisas, oferecendo uma solução prática e eficiente para melhorar a interoperabilidade e integração de dispositivos IoT.

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

Abstract

SURNAME, FirstName MiddleName1 MiddleName2. **Work title**: work subtitle. Julho de 2023. 39 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, DefenseYear.

Write here the English version of your "Resumo". Example text, example text.

Keywords: Keyword1. Keyword2. Keyword3. etc.

Lista de figuras

Sumário

1 Introdução

Neste capítulo, serão apresentados a contextualização, a justificativa/problema de pesquisa, a hipótese/proposição, os objetivos e o método de pesquisa do projeto. Além disso, será feita uma breve visão geral da estrutura do documento.

1.1 Contextualização e Motivação

Este projeto de pesquisa está inserido na área de Internet das Coisas (IoT), que se refere à interconexão de dispositivos inteligentes em uma rede global. A IoT tem o potencial de transformar diversos setores, como saúde, indústria, agricultura e cidades inteligentes, trazendo benefícios como automação, eficiência energética e monitoramento remoto.

No entanto, a interoperabilidade entre dispositivos IoT de diferentes fabricantes ainda é um desafio significativo. A falta de padronização e a diversidade de protocolos e interfaces dificultam a comunicação e integração entre esses dispositivos. Isso limita a capacidade de criar soluções abrangentes e escaláveis que aproveitem todo o potencial da IoT.

Diante desse cenário, o projeto de pesquisa propõe abordar o problema da interoperabilidade, focando na implementação e melhoria do WoTPy, um *gateway* experimental baseado no W3C-WoT. O objetivo é desenvolver uma solução que facilite a comunicação e a integração entre dispositivos IoT de diferentes fabricantes, seguindo os padrões e diretrizes estabelecidos pelo W3C.

1.2 Justificativa/Problema de Pesquisa

A justificativa para a realização deste projeto de pesquisa reside na necessidade de superar as barreiras de interoperabilidade na IoT. A falta de padronização e a diversidade de protocolos e interfaces dificultam a comunicação e a integração entre os dispositivos IoT, limitando o potencial da tecnologia.

O problema de pesquisa consiste em desenvolver uma solução que promova a interoperabilidade entre dispositivos IoT de diferentes fabricantes. O objetivo é permitir

que esses dispositivos se comuniquem e interajam de forma transparente, possibilitando a criação de soluções abrangentes e escaláveis na IoT.

1.3 Hipótese/Proposição

Com base na análise do problema de pesquisa, é proposta a hipótese de que a implementação e melhoria do WoTPy como um *gateway* experimental baseado no W3C-WoT pode facilitar a comunicação e a integração entre dispositivos IoT de diferentes fabricantes. Acredita-se que essa solução, alinhada aos padrões e diretrizes estabelecidos pelo W3C, pode contribuir para a superação das barreiras de interoperabilidade na IoT.

1.4 Objetivos

O objetivo geral deste projeto de pesquisa é desenvolver e aprimorar o WoTPy como um *gateway* experimental baseado no W3C-WoT, visando melhorar a interoperabilidade entre dispositivos IoT de diferentes fabricantes. Para atingir esse objetivo geral, foram estabelecidos os seguintes objetivos específicos:

- Compreender as especificações do W3C-WoT e suas principais componentes, como Descrição das Coisas (*Thing Descriptions*), Modelos de Vinculação (*Binding Templates*) e *Scripting API*.
- Selecionar e estudar as bibliotecas e ferramentas necessárias para o desenvolvimento do WoTPy.
- Resolver problemas de instalação e configuração do WoTPy, visando facilitar sua implantação em diferentes ambientes.
- Realizar testes e validações para garantir a funcionalidade e qualidade do WoTPy.
- Desenvolver exemplos de uso práticos do WoTPy e documentação detalhada para auxiliar desenvolvedores e usuários na implementação e utilização do *gateway*.

1.5 Método de Pesquisa

Este projeto de pesquisa adota uma abordagem 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 especificações do W3C-WoT e estudo aprofundado de suas principais componentes.
3. Seleção e estudo das bibliotecas e ferramentas relacionadas ao WoTPy.
4. Desenvolvimento e aprimoramento do WoTPy, incluindo resolução de problemas de instalação, testes e validações.
5. Criação de exemplos de uso práticos do WoTPy e elaboração de documentação detalhada.

1.6 Estrutura do Documento

Este documento está estruturado da seguinte forma:

- Capítulo 1: Introdução - apresenta a contextualização, a justificativa, a hipótese, os objetivos e o método de pesquisa do projeto.
- Capítulo 2: Objetivos - apresenta 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 no desenvolvimento do projeto.
- Capítulo 5: Resultados - apresenta os resultados obtidos durante a implementação 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 abordados no trabalho, apresenta as conclusões alcançadas e sugere possíveis direções futuras.
- Apêndice A: relata as dificuldades e os problemas enfrentados durante o desenvolvimento do trabalho.

2 Objetivos

2.1 Objetivo Geral

o objetivo geral deste trabalho foi contribuir para o desenvolvimento do WoTPy Mangas *et al.* (2022), um *gateway* experimental baseado no W3C-WoT, que melhora a interoperabilidade entre dispositivos IoT de diferentes fabricantes, facilitando a comunicação e integração em uma única aplicação ou sistema.

2.2 Objetivo Específico

para atingir o objetivo geral, os seguintes objetivos específicos foram estabelecidos:

1. Analisar as especificações do W3C-WoT Lagally *et al.* (2023) e compreender os principais conceitos e requisitos para a implementação do WoTPy;
2. Selecionar e estudar as bibliotecas e ferramentas das dependências do WoTPy;
3. Resolver problemas de instalação, garantindo que o WoTPy possa ser facilmente implantado e configurado em diferentes ambientes e sistemas;
4. Testar e validar as contribuições feitas;
5. Desenvolver exemplo de uso e documentação detalhada para auxiliar desenvolvedores e usuários na implementação do WoTPy em seus projetos IoT;

2.3 Extensão

com esses objetivos específicos alcançados no primeiro semestre de 2023, pretende-se abordar, no segundo semestre de 2023, a integração do wotpy com grafos de conhecimento. Essa integração busca aprimorar ainda mais a interoperabilidade dos dispositivos IoT, possibilitando a divulgação das capacidades dos sensores e suas observações por meio de um *endpoint SPARQL*.

3 Revisão Bibliográfica

Neste capítulo, será apresentada uma revisão bibliográfica sobre a Web das Coisas (WoT) e sua arquitetura proposta pelo World Wide Web Consortium (W3C). Também serão discutidos os desafios relacionados à interoperabilidade entre redes IoT e o papel dos *gateways* nesse contexto.

3.1 Web das Coisas e a Arquitetura Proposta pelo W3C

A Web das Coisas (WoT) é um conceito que tem ganhado destaque nos últimos anos. O W3C define a WoT como a interconexão de sensores, atuadores, objetos físicos, locais e até mesmo pessoas por meio da Internet. O objetivo da WoT é utilizar as tecnologias Web para facilitar o desenvolvimento de aplicações e serviços que envolvam esses dispositivos e suas representações virtuais (GROUP, 2015b; GROUP, 2015a).

A arquitetura proposta pelo W3C para a WoT define os elementos que compõem essa rede interconectada. Ela destaca a importância da infraestrutura e dos protocolos da Internet para viabilizar a comunicação entre os dispositivos. Além disso, a arquitetura enfatiza a existência de dispositivos de borda, também conhecidos como *gateways*, que desempenham um papel fundamental na interoperabilidade entre as redes IoT e a *Web* (MATSUKURA *et al.*, 2023).

A arquitetura da WoT define os dispositivos de borda como pontos de interconexão entre a rede interna e a *Internet*. Esses dispositivos são responsáveis por realizar a computação e o processamento de dados próximo aos sensores e atuadores, muitas vezes em ambientes com recursos limitados. Essa abordagem, conhecida 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 Arcabouço WoTPy

No contexto da WoT, o WoTPy se destaca como um arcabouço desenvolvido no ambiente acadêmico com o objetivo de implementar as recomendações do W3C. O WotPy

é um conjunto de ferramentas que permite o desenvolvimento de *gateways* IoT, abrangendo diferentes protocolos de comunicação, como HTTP, MQTT e CoAP. Ele foi projetado para simplificar a interoperabilidade entre dispositivos e oferecer suporte a funcionalidades essenciais, como privacidade, segurança, descoberta de dispositivos e interfaces de usuário (García Mangas; Suárez Alonso, 2019).

4 Metodologia

A metodologia utilizada para a realização deste trabalho, apresentará as etapas e abordagens adotadas. A metodologia foi planejada com o objetivo de alcançar os resultados esperados e responder às questões de pesquisa estabelecidas.

4.1 Abordagem de Pesquisa

A abordagem de pesquisa adotada neste trabalho foi baseada em uma combinação de pesquisa exploratória, pesquisa bibliográfica e desenvolvimento prático. A pesquisa exploratória foi realizada para obter uma compreensão aprofundada do tema, identificar os principais conceitos e tendências, e explorar as soluções existentes. A pesquisa bibliográfica foi conduzida para revisar e analisar as principais publicações, artigos científicos e padrões relacionados ao WoTPy e ao W3C-WoT. Isso permitiu embasar teoricamente o trabalho e identificar lacunas ou oportunidades de melhoria. O desenvolvimento prático envolveu a implementação e o teste do WoTPy e a criação de exemplos de uso e documentação detalhada.

4.2 Coleta de Dados

A coleta de dados foi realizada por meio de diversas fontes, incluindo artigos científicos, publicações, documentação oficial do W3C-WoT e WoTPy, bem como experimentações práticas. A revisão bibliográfica foi conduzida para coletar informações relevantes sobre o W3C-WoT, as tecnologias envolvidas, os padrões e as melhores práticas. A documentação oficial do W3C-WoT Lagally *et al.* (2023) e do WoTPy Mangas *et al.* (2018) foi consultada para compreender as especificações, APIs e diretrizes recomendadas.

4.3 Desenvolvimento do WoTPy

O desenvolvimento do WoTPy foi conduzido com base nas especificações e diretrizes do W3C-WoT. Inicialmente, foi realizada uma análise detalhada das especificações do W3C-WoT para compreender os principais conceitos e requisitos. Em seguida, foram selecionadas as bibliotecas e ferramentas adequadas como dependências do WoTPy. Foram

resolvidos problemas de instalação e configuração para garantir a facilidade de implantação em diferentes ambientes e sistemas operacionais. A implementação do WoTPy foi realizada utilizando a linguagem de programação Python e seguindo as boas práticas de desenvolvimento de *software*.

4.4 Validação e Testes

A validação do WoTPy foi realizada por meio de um conjunto de testes disponíveis no repositório wot-py Mangas *et al.* (2022). Os testes foram projetados para verificar a funcionalidade, a compatibilidade e a interoperabilidade do WoTPy. Foram realizados testes unitários, testes de integração e testes de sistema para garantir a qualidade e a robustez do WoTPy. Os resultados dos testes foram analisados e registrados para avaliar o desempenho e identificar possíveis problemas ou melhorias.

4.5 Desenvolvimento do Exemplo de Uso e Documentação

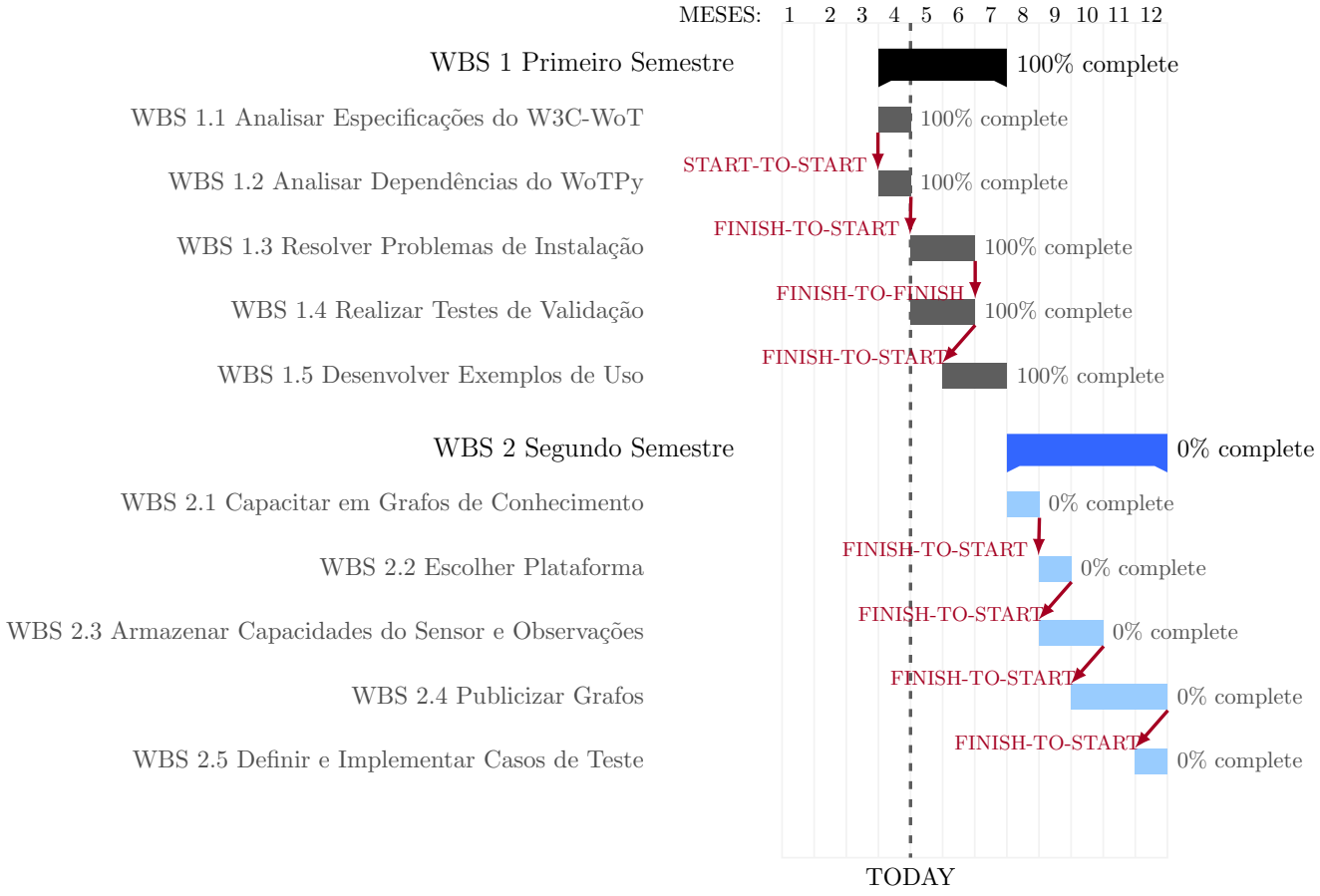
Para auxiliar os desenvolvedores e usuários na implementação do WoTPy em seus projetos IoT, foi desenvolvido um exemplo de uso práticos e documentação detalhada. O exemplo de uso abrangem cenários comuns de aplicação do WoTPy, demonstrando como utilizar suas funcionalidades e integrar dispositivos IoT. A documentação detalhada fornece informações sobre a instalação, configuração, e melhores práticas para aproveitar ao máximo o WoTPy.

4.6 Cronograma

O trabalho foi conduzido em etapas sequenciais, com um cronograma definido para orientar o progresso e a organização das atividades, como demonstrado na figura 1.

Cada etapa foi planejada para ser concluída dentro de um período de tempo específico, permitindo o avanço contínuo do trabalho e o cumprimento dos prazos estabelecidos.

Figura 1 – Cronogrma



5 Resultados

Os resultados esperados, descritos no plano de atividades Ariga e Nakano (2023), foram alcançados de forma satisfatória durante o desenvolvimento do trabalho. A seguir, são apresentados os resultados obtidos:

Compreensão detalhada das especificações do W3C-WoT: Foi realizada uma análise abrangente das especificações do W3C-WoT, resultando em uma compreensão aprofundada dos principais conceitos e requisitos para a implementação do WoTPy. Isso permitiu a correta e eficiente implementação do *gateway*, seguindo os padrões estabelecidos pelo consórcio.

Seleção e domínio das bibliotecas e ferramentas adequadas: Foram selecionadas as bibliotecas e ferramentas mais adequadas para o desenvolvimento do WoTPy, levando em consideração a sua compatibilidade com os protocolos de comunicação necessários e a facilidade de integração com outros projetos IoT. O domínio dessas ferramentas foi crucial para o sucesso do trabalho.

Resolução dos problemas de instalação do WoTPy: Foram identificados e solucionados os problemas de instalação do WoTPy, resultando em um processo simplificado e eficiente para implantar e configurar o *gateway* em diferentes ambientes e sistemas operacionais. Isso facilitou a adoção do WoTPy por parte dos desenvolvedores e usuários finais, garantindo uma experiência mais fluida.

Validação das contribuições por meio de testes: Um conjunto abrangente de testes foi realizado para validar as contribuições feitas ao WoTPy. Esses testes garantiram a qualidade e a funcionalidade do *gateway*, verificando sua conformidade com os requisitos e sua capacidade de lidar com diferentes casos de uso.

Desenvolvimento de exemplos de uso e documentação detalhada: Foi criado um exemplo prático de uso do WoTPy, demonstrando sua aplicabilidade em cenário de IoT. Além disso, uma documentação detalhada foi elaborada, fornecendo orientações claras e recursos auxiliares para facilitar a implementação do WoTPy em projetos IoT. Isso contribuiu para a disseminação e adoção do *gateway* pela comunidade.

5.1 Especificações do W3C-WoT

O Web das Coias (WoT) é uma iniciativa do *World Wide Web Consortium* (W3C) que busca estabelecer uma estrutura para conectar dispositivos e serviços na Internet das Coisas (IoT). O objetivo do WoT é permitir a interoperabilidade entre esses dispositivos, facilitando a comunicação e a integração padronizadas.

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

- Descrição de Coisas (*Thing Description*) Kaebisch *et al.* (2020) define um formato de dados legível por máquina 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) oferecem orientações sobre como definir interfaces de rede em Coisas para protocolos específicos e ecossistemas de IoT. Esses modelos são úteis para garantir a compatibilidade e a integração de diferentes sistemas.
- Descoberta (*Discovery*) Cimmino *et al.* (2020) define um mecanismo de distribuição de metadados das Coisas. Ela permite a localização e o acesso a informações detalhadas sobre as Coisas, facilitando a descoberta e a integração de dispositivos na rede.
- Modelos de Vinculação (*Scripting API*) Kis *et al.* (2020) possibilita 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, é um formato de dados legível por máquina que descreve os metadados e as interfaces de rede das Coisas na Web das Coisas. Essa especificação, desenvolvida pelo W3C, fornece uma base sólida para promover a interoperabilidade entre as Things e a Web.

Através do WoT Thing Description, é possível descrever as características, propriedades, funcionalidades e interfaces de uma Thing de forma estruturada e padronizada. Essa descrição inclui informações detalhadas sobre como interagir com a Thing, como acessar seus recursos e como interpretar os dados que ela oferece.

Ao utilizar o formato de Descrição de Coisas WoT, torna-se mais fácil para diferentes dispositivos e sistemas compreenderem as capacidades e os requisitos de uma Thing específica. Isso possibilita a criação de aplicações e serviços que possam interagir de forma transparente com uma ampla variedade de Things, independentemente do fabricante ou da tecnologia subjacente.

A interoperabilidade é um aspecto fundamental na Web das Coisas, pois permite que as Things se comuniquem e interajam de maneira eficiente, facilitando a integração e a criação de soluções inovadoras. O WoT Thing Description desempenha um papel crucial nesse contexto, ao estabelecer uma linguagem comum para descrever as características das Things e ao promover a padronização na forma como essas informações são compartilhadas.

5.1.2 Modelos de Vinculação

Os Modelos de Vinculação, são diretrizes que oferecem orientações sobre como definir interfaces de rede em Coisas para protocolos específicos e ecossistemas de IoT. Esses modelos são extremamente úteis para garantir a compatibilidade e a integração entre diferentes sistemas.

Cada protocolo de comunicação utilizado na Web das Coisas possui suas próprias especificidades e requisitos. Os WoT Binding Templates fornecem um conjunto de instruções e recomendações para mapear as funcionalidades e os recursos de uma Thing em um formato compatível com um determinado protocolo.

Ao utilizar os WoT Binding Templates, os desenvolvedores podem criar interfaces de rede para suas Things que são consistentes com as exigências e os padrões dos protocolos específicos. Isso facilita a comunicação e a interação entre as Things e os sistemas que utilizam esses protocolos, garantindo uma maior interoperabilidade e uma integração mais suave.

Além disso, os WoT Binding Templates ajudam a promover a reutilização de código e a padronização na implementação das interfaces de rede. Ao seguir essas orientações, os desenvolvedores podem garantir que suas Things sejam facilmente integradas a ecossistemas de IoT existentes, evitando a necessidade de desenvolver soluções personalizadas para cada protocolo.

A utilização dos WoT Binding Templates também contribui para a escalabilidade e a flexibilidade dos sistemas de IoT. Como esses modelos oferecem uma estrutura padronizada para a definição de interfaces de rede, é mais fácil adicionar novas Things ao ecossistema e integrá-las com os sistemas existentes. Isso promove a expansão dos ecossistemas de IoT e facilita o desenvolvimento de soluções mais abrangentes e interconectadas.

5.1.3 Descoberta WoT

A Descoberta WoT, é um mecanismo que define a distribuição de metadados das Things na Web das Coisas. Essa especificação, desenvolvida pelo W3C, permite a localização e o acesso a informações detalhadas sobre as Things, facilitando a descoberta e a integração de dispositivos na rede.

A Descoberta WoT desempenha um papel fundamental na Web das Coisas, pois permite que os dispositivos IoT sejam encontrados e identificados de forma eficiente. Isso é especialmente importante em ambientes onde há uma grande quantidade de Things interconectadas, tornando a descoberta manual desses dispositivos inviável.

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

Além disso, a Descoberta WoT possibilita a integração de dispositivos de diferentes fabricantes e tecnologias. Por meio da distribuição de metadados padronizados, os sistemas de IoT podem identificar e interagir com as Things de forma consistente, independentemente de suas características individuais.

A Descoberta WoT também contribui para a escalabilidade e a flexibilidade dos sistemas de IoT. Com a capacidade de localizar e acessar informações detalhadas sobre as Things, é mais fácil adicionar novos dispositivos à rede e integrá-los com os sistemas existentes. Isso promove a expansão dos ecossistemas de IoT e facilita o desenvolvimento de soluções mais abrangentes e interconectadas.

5.1.4 API de Scripting WoT

A API de Scripting WoT, é uma especificação que possibilita a implementação da lógica de aplicação das Things utilizando uma API JavaScript comum. Essa abordagem simplifica o desenvolvimento de aplicativos IoT e promove a portabilidade entre fornecedores e dispositivos.

Ao utilizar a WoT Scripting API, os desenvolvedores podem escrever o código de lógica de aplicação das Things em JavaScript, uma linguagem amplamente adotada e conhecida por sua flexibilidade e facilidade de uso. Isso elimina a necessidade de aprender linguagens de programação específicas de cada dispositivo ou fornecedor, tornando o processo de desenvolvimento mais eficiente e acessível.

A API de Scripting WoT fornece um conjunto de funcionalidades e métodos que permitem interagir com as Things, acessar seus recursos, enviar comandos e receber dados. Essa abordagem simplifica a implementação da lógica de negócios e a integração com outros componentes do sistema.

Além disso, a utilização da WoT Scripting API promove a portabilidade entre fornecedores e dispositivos IoT. Como a API é baseada em JavaScript, um padrão amplamente suportado, os aplicativos desenvolvidos com essa abordagem podem ser executados em diferentes dispositivos e plataformas, independentemente do fabricante.

Essa portabilidade é especialmente valiosa em cenários de IoT, onde há uma diversidade de dispositivos e fornecedores. Com a WoT Scripting API, é possível de-

senvolver aplicativos que podem ser implantados em uma ampla variedade de dispositivos IoT, reduzindo a dependência de fornecedores específicos e oferecendo mais flexibilidade aos desenvolvedores.

5.1.5 Diretrizes de Segurança e Privacidade WoT

As Diretrizes de Segurança e Privacidade WoT, são um conjunto de orientações que oferecem diretrizes para a implementação segura das Things e abordam questões relacionadas à segurança e privacidade nos sistemas WoT. Essas diretrizes são de extrema importância para proteger os dispositivos e os dados sensíveis envolvidos nas redes WoT.

A segurança e a privacidade são considerações críticas na Web das Coisas, pois os dispositivos IoT estão cada vez mais presentes em nosso cotidiano e lidam com uma grande quantidade de dados sensíveis. As WoT Security and Privacy Guidelines fornecem recomendações e melhores práticas para garantir a integridade, confidencialidade e disponibilidade dos dispositivos e dos dados na rede WoT.

Essas diretrizes abrangem uma variedade de aspectos de segurança, incluindo autenticação, autorização, criptografia, gerenciamento de chaves, proteção contra ataques e privacidade dos dados. Elas ajudam os desenvolvedores a identificar e implementar medidas de segurança apropriadas em suas Things, mitigando os riscos de exposição a ameaças e violações de privacidade.

Ao seguir as WoT Security and Privacy Guidelines, os desenvolvedores podem adotar práticas de segurança eficazes desde o design até a implementação das Things. Isso inclui a utilização de criptografia adequada para proteger a comunicação entre as Things, a implementação de mecanismos de autenticação robustos para garantir a identidade dos dispositivos e a adoção de políticas de privacidade que respeitem os direitos dos usuários.

Além disso, as diretrizes também incentivam a manutenção contínua da segurança e privacidade dos dispositivos e sistemas WoT, por meio de monitoramento, atualizações de segurança e resposta efetiva a incidentes. Isso é fundamental para garantir que os dispositivos IoT permaneçam protegidos ao longo do tempo, considerando a constante evolução das ameaças cibernéticas.

5.2 Dependências do W3C-WoTPy

O arquivo ”setup.py” Ariga (2023c) é responsável por definir as dependências necessárias para a execução da biblioteca WotPy, conforme especificado pelo *World Wide Web Consortium* (LAGALLY *et al.*, 2023). Essas dependências podem ser agrupadas em duas categorias: as obrigatórias e as opcionais.

Entre as principais dependências obrigatórias definidas no arquivo setup.py, estão:

- tornado: uma biblioteca assíncrona utilizada para criar aplicativos web em Python. É empregada pela WotPy para estabelecer servidores HTTP e WebSocket.
- jsonschema: uma biblioteca que valida esquemas JSON e os dados JSON correspondentes. A WotPy utiliza essa biblioteca para validar as descrições de coisa (Thing Descriptions) recebidas e geradas.
- six: uma biblioteca que possibilita escrever código Python compatível tanto com a versão 2 quanto com a versão 3. A WotPy se beneficia dessa biblioteca para garantir a compatibilidade entre as duas versões do Python.
- rx: uma biblioteca de programação reativa que permite escrever 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 converte strings para ”slug”, um formato de texto que utiliza apenas caracteres ASCII, números e traços. A WotPy utiliza essa biblioteca para criar identificadores únicos para as coisas.

Além dessas dependências obrigatórias, existem algumas dependências opcionais que a WotPy utiliza se estiverem disponíveis 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 para implementar o protocolo Message Queue Telemetry Transport (MQTT), utilizada pela WotPy para suportar o protocolo MQTT.

- websockets: uma biblioteca Python para suportar a comunicação via WebSocket, utilizada 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.

Essas dependências são verificadas em tempo de execução e adicionadas ao conjunto de dependências da biblioteca WotPy, caso estejam disponíveis no sistema.

5.3 Resolução dos Problemas e Validação

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

5.3.1 Construção do Projeto com Docker

Ao tentar construir o projeto utilizando o "*docker build .*", deparou-se com um erro relacionado à versão do "pacote *numpy*". A mensagem de erro está descrito neste arquivo (ARIGA, 2023d).

Para solucionar esse problema, foi tentada inicialmente a alteração da versão do Python no arquivo "*Dockerfile*" (ARIGA, 2023f).

Posteriormente, optou-se por utilizar 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, 2023g).

Por fim, a solução adotada foi remover a inclusão das dependências do diretório "/examples/benchmark" no "*Dockerfile*", mantendo a modificação anterior (ARIGA, 2023h). Essa 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 Execução dos Testes

A execução dos testes do WoTPy foi iniciada com o seguinte comando"*./pytest-docker-all.sh*". Após a construção, foi identificado o erro descrito neste arquivo Ariga (2023e). Para lidar com esse problema, o arquivo setup.py foi editado (ARIGA, 2023f).

Dessa forma, o WoTPy pôde ser construído corretamente utilizando o Docker e passou nos testes propostos no "pytest-docker-all.sh'.

5.4 Desenvolvimento do Exemplo de Uso

O exemplo demonstra a transmissão de dados do sensor UV de um microcontrolador ESP32 para um servidor Web of Things (WoT) via protocolo HTTP. O projeto utiliza a biblioteca WoTPy para criação do servidor e interações com o WoT, e o microcontrolador ESP32 emparelhado com um sensor UV ML8511.

Dois principais arquivos de código compõem este projeto: server.py e main.py. O arquivo server.py configura o servidor WoT, expõe um Thing que representa o sensor UV, e define *handlers* customizados para leitura e gravação dos dados do sensor UV. O main.py, rodando no ESP32, periodicamente lê os dados do sensor UV e os envia para o servidor WoT usando requisições HTTP.

Este projeto serve como um exemplo base de integração de dispositivos IoT com princípios do WoT, facilitando a comunicação e a interoperabilidade entre dispositivos e aplicações dentro de um ecossistema de IoT.

5.4.1 Protocolo de Comunicação

A situação envolve um dispositivo ESP32 equipado com um sensor ML8511 (sensor UV) atuando como cliente, enquanto um computador atua como servidor utilizando a biblioteca WoTPy.

A comunicação entre o cliente ESP32 e o servidor Web of Things (WoT) é estabelecida por meio do protocolo HTTP, sendo uma escolha adequada para esse contexto. O uso do protocolo HTTP apresenta diversas vantagens que tornam essa abordagem viável e eficiente para o projeto em questão.

Uma das razões para a escolha do protocolo HTTP é a familiaridade do autor com essa tecnologia. Essa familiaridade agiliza o processo de desenvolvimento, evitando a necessidade de aprender um novo protocolo e me permitindo aproveitar a experiência prévia nessa área.

Além disso, o HTTP é amplamente utilizado na web e conta com recursos de implementação disponíveis para diferentes plataformas e linguagens de programação, incluindo o *MicroPython* utilizado no ESP32. Existem bibliotecas e ferramentas prontamente disponíveis que facilitam a implementação da comunicação HTTP, possibilitando um desenvolvimento mais eficiente e eficaz.

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

5.4.2 Cliente Web of Thing (WoT)

O código ”main.py” Ariga (2023a) implementa um cliente Web of Things (WoT) no dispositivo ESP32, permitindo a interação com sensores e o envio dos dados para um servidor WoT. O cliente WoT é responsável por buscar informações dos sensores e controlar as funcionalidades dos dispositivos conectados de maneira padronizada e interoperável.

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

O código define os sensores utilizados no dispositivo ESP32, como o sensor de UV, e suas configurações. Cada sensor possui uma identificação única e é associado a uma descrição no formato de Thing Description (TD). A TD contém informações sobre o sensor, como os links de acesso aos dados no servidor WoT.

A função *send_sensor_data* é responsável por enviar os dados dos sensores para o servidor WoT. Ela recebe como parâmetros a identificação do sensor, o tipo de sensor, o URL de acesso aos dados na TD 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 falha é exibida juntamente 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 utilizando a função *send_sensor_data*. Esse processo permite que os dados dos sensores sejam atualizados no servidor WoT em tempo real, possibilitando que os usuários acessem e utilizem essas informações de forma padronizada, como é representado na figura 2.

5.4.3 Servidor Web of Things (WoT)

O servidor Web of Things (WoT) é uma peça fundamental na arquitetura WoT, pois é responsável por expor os dispositivos conectados e suas funcionalidades para que possam ser descobertos e interagidos pelos clientes. O servidor WoT atua como uma ponte entre os dispositivos da Internet das Coisas (IoT) e os aplicativos e serviços que desejam acessá-los.

O código server.py Ariga (2023b) implementa um servidor Web of Things (WoT) que expõe uma Thing (dispositivo) responsável por fornecer os valores de um sensor de UV. O servidor WoT permite a descoberta e interação com a Thing por meio de solicitações HTTP padronizadas.

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

No código, é criado um HTTP server na porta especificada (HTTP_PORT) e um Servient, que é uma instância responsável por gerenciar os recursos WoT. A descrição

da Thing é definida, contendo o identificador (ID_THING) e as propriedades, como o sensor de UV.

Para cada propriedade da Thing, como o sensor de UV, são definidos os manipuladores de leitura (read_uv) e escrita (write_uv). O manipulador de leitura é responsável por retornar o valor atual do sensor quando solicitado pelo cliente. O manipulador de escrita é responsável por atualizar o valor do sensor quando recebido uma solicitação de escrita do cliente.

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

Por fim, a Thing é exposta e o servidor inicia seu loop de eventos, aguardando as solicitações dos clientes e respondendo de acordo com as interações definidas, como é representado na figura 2.

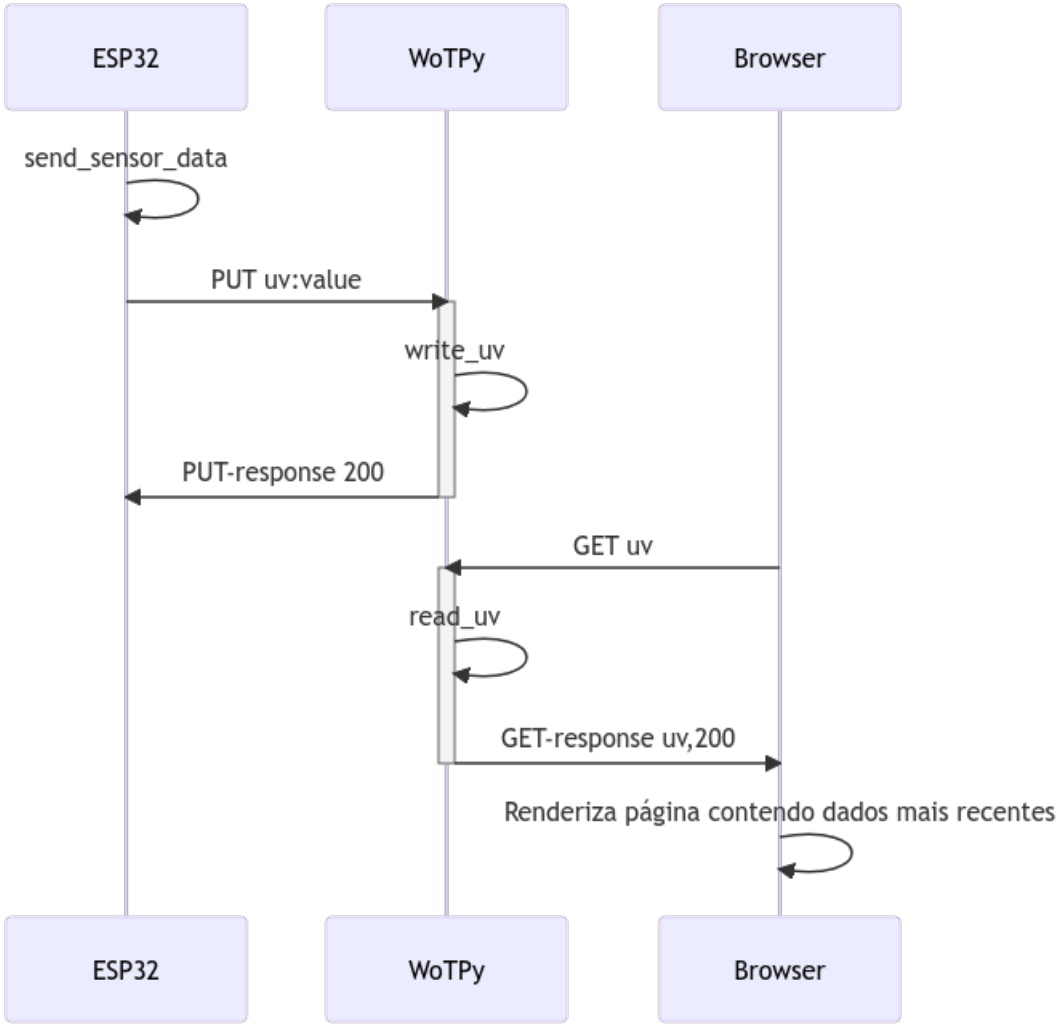


Figura 2 – O dispositivo (ESP32) executa um loop que, a cada 5 segundos executa a função send_sensor_data. Esta função envia uma requisição PUT para o servidor (WoTPy). Em resposta a essa requisição o servidor executa o handler (função) write_uv que contém o envio da resposta. Um usuário pode acessar 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 28.

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 29.

ARIGA, G. T. *Arquivo setup.py*. 2023. Disponível em: ⟨https://github.com/T16K/wot-py/blob/develop/setup.py⟩. Citado na página 25.

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 26.

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 27.

ARIGA, G. T. *WoTPy Versão 1.1*. 2023. Disponível em: ⟨https://github.com/agmangas/wot-py/commit/b8e9d090e738b343a825cd404f63dde950954a70⟩. Citado 2 vezes nas páginas 26 e 27.

ARIGA, G. T. *WoTPy Versão 1.2*. 2023. Disponível em: ⟨https://github.com/agmangas/wot-py/commit/61bec7348c6342c05fd8d2c3ccb21dad60aed58b⟩. Citado na página 26.

ARIGA, G. T. *WoTPy Versão 1.5*. 2023. Disponível em: ⟨https://github.com/agmangas/wot-py/commit/3dd4f7e428bd565970adf96f8f4482cf986496ea⟩. Citado na página 26.

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 26.

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 25.

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 26.

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 26.

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.

Apêndice A – Problemas e Dificuldades