UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE COMPUTAÇÃO

Weverton Carlos Leal de Oliveira

PLATAFORMA DE SERVIÇOS PARA INTERNET DAS COISAS

ALEGRE – ES
2018

Weverton Carlos Leal de Oliveira

PLATAFORMA DE SERVIÇOS PARA INTERNET DAS COISAS

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, sob orientação do Professor Dr. Helder Mendes Amorim.

ALEGRE – ES

2018

PLATAFORMA DE SERVIÇOS PARA INTERNET DAS COISAS

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 21 de Junho de 2018.

COMISSÃO EXAMINADORA

Prof. Dr. Helder de Amorim Mendes
Universidade Federal do Espírito Santo

Prof. Dr. Clayton Vieira Fraga Filho
Universidade Federal do Espírito Santo

Prof. M. Sc. Rodrigo Freitas Silva
Universidade Federal do Espírito Santo

Dedico todo este trabalho aos meus pais, Elza Leal e José Carlos, e a meu irmão Wenderson Leal, por todo incentivo e apoio que me ofereceram, não só nos momentos críticos, mas durante esta longa caminhada.

AGRADECIMENTOS

Primeiramente, agradeço a Deus que permitiu que tudo isso acontecesse, me dando saúde mental e física para superar as dificuldades e por tudo o que Ele tem feito em minha, sendo Ele o maior mestre que alguém pode conhecer.

Agradeço aos meus pais, ao meu irmão e minha companheira, pelo amor incondicional, que mesmo nos momentos de minha ausência dedicados aos estudos me incentivaram nas horas que mais precisei, sem eles não seria possível a realização deste trabalho.

Ao Alojamento Estudantil que serviu não só como morada, mas como base de para meu aprendizado e todos os moradores ali presentes e os que já passaram por lá, com quem compartilhei momentos de alegria, tristeza e ansiedade, sendo parte da minha formação e que vão continuar presentes em minha vida com certeza.

Agradeço a meu orientador Helder pela oportunidade e suporte na elaboração deste trabalho, por sua paciência, correções e incentivos.

E por último, mas não menos importante agradeço aos demais professores do departamento por todo o conhecimento transmitido, servindo como apoio para elaboração deste trabalho.

RESUMO

No decorrer dos anos, podemos observar grandes evoluções técnicas e cientificas, onde o hardware está ficando menor, com maior capacidade de processamento e relativamente mais barato, onde novas aplicações complexas são baseadas na composição e colaboração com outros serviços. Para se tirar proveito dessas tendências, surgiu a Internet das Coisas (*Internet of Things* (IoT)) que vem crescendo exponencialmente, assim como a idealização de projetos que utilizam essa tecnologia. O conceito de IoT, se refere a uma revolução tecnológica que possui como objetivo a conexão entre os objetos usados no dia a dia através da rede mundial de computadores. A conexão com a internet viabiliza gerir remotamente os objetos, permitindo que os objetos sejam acessados como provedores de serviços. A adoção de uma plataforma para IoT visa facilitar a construção de aplicações para IoT. Tais plataformas são responsáveis por processar informações e prove um meio padronizado para o acesso aos dados armazenados em servidores (*WebService*). Este trabalho propõe uma plataforma de serviços para dispositivos IoT. Esta plataforma irá integrar e gerenciar as funcionalidades individuais de dispositivos de uma rede IoT, que irá permitir a criação de novas funcionalidades especificas para o cliente.

Palavras-chave: Plataformas para IoT, Internet das Coisas, Academic-IoT.

LISTA DE DIAGRAMAS

Diagrama 1 - Diagrama de Caso de Uso 35
Diagrama 2 - Modelo Entidade Relacionamento 38
Diagrama 3 - Diagrama de sequencia.....40

LISTA DE FIGURAS

Figura 1 - Tela inicial 41
Figura 2 - Tela de login 42
Figura 3 - Recupera senha 42
Figura 4 - Tela cadastro de usuário 43
Figura 5 - Tela principal 43
Figura 6 - Tela principal: Informação dasboard 44
Figura 7 - Tela de cadastro de dashboard 45
Figura 8 - Tela principal: Informações do projeto. 45
Figura 9 - Tela Principal: Informações do projeto2. 46
Figura 10 - Tela Principal: Informações do projeto3 46
Figura 11 - Tela cadastro de projeto 47
Figura 12 - Tela Principal : Informações do recurso. 47
Figura 13 - Tela de cadastro de recurso 48
Figura 14 - Tela de perfil do usuário 49

LISTA DE TABELAS

Tabela 1 - Tabela de Requisitos versus Plataformas IoT 27

LISTA DE SIGLAS

Ajax - *Asynchronous Javascript and XML.*

API - *Application Programming Interface.*

CoAP - *Constrained Application Protocol.*

HTTP - *HyperText Transfer Protocol.*

IDE - *Integrated Development Environment.*

IPv6 - *Internet Protocol version 6*.

JDBC - *Java Database Connectivity.*

JPA - Java *Persistence* API.

JSON - *JavaScript Object Notation*.

JSP - *JavaServer Pages.*

JSTL - *Java Server Pages Standard Tag Library.*

M2M - Machine to Machine.

MER - Modelo Entidade Relacionamento.

MVC - *Model, Controller* e *View.*

QoS - *Quality of Service.*

REST - *Representational State Transfer.*

RFID - *Radio-Frequency Identification.*

SGBD - Sistema Gerenciador de Banco de Dados.

SOAP - *Simple Object Access Protocol.*

TCP - *Transmission Control Protocol.*

UML - *Unified Modeling Language.*

UC - *User Case.*

UPnP - *Universal Plug and Play.*

URI - *Uniform Resource Identifier.*

XML - *eXtensible Markup Language.*

Sumário

Lista de Abreviaturas/Siglas 9
1. Introdução 12
1.1 Objetivo 14
1.2 Objetivos específicos 14
2. Revisão Bibliográfica 15
1.3 Internet das Coisas – IoT (*Internet of Things*) 15
1.3.1 História 15
1.3.2 Evolução 16
1.3.3 Componentes básicos 17
1.3.4 Desafios 17
1.3.5 Principal problema 18
1.4 Ipv6 18
1.5 Plataformas 19
1.5.1 Objetivos das plataformas de IoT 20
1.5.2 Exemplos de plataformas para Internet das Coisas 21
1.5.3 Comparação entre as plataformas existentes 25
1.6 Protocolos para Internet das Coisas 28
1.6.1 CoAP 28
1.6.2 UPnP 29
1.6.3 MQTT 30
3. Metodologia 32
4. Resultados 35
4.1 Diagrama de caso de uso 35
4.2 Diagrama Entidade Relacionamento (MER) 38
4.3 Diagrama de Sequencia 40
4.4 Telas 41
4.4.1 Tela Inicial 41
4.4.2 Tela Login 41
4.4.3 Tela Principal 43
5. Conclusão 50
6. Referências Bibliográfica 51
7. ANEXOS 55

1. Introdução

Com o passar dos anos, tem-se observado grandes evoluções tecnológicas no que diz respeito ao mundo do dispositivo, onde o hardware está ficando mais barato, menor em tamanho e com maior capacidade processamento e/ou armazenamento. Outra grande evolução vem do domínio de *software* de negócios, onde novas aplicações complexas são baseadas na composição e colaboração com outros serviços (GUINARD et al., 2010). Para se tirar proveito dessas duas tendências, a IoT vem crescendo exponencialmente, assim como a idealização de projetos que utilizam essa tecnologia.

O conceito de "Internet das Coisas" entende-se como uma revolução tecnológica que possui como objetivo conectar os objetos usados no dia a dia através da rede mundial de computadores (SILVA et al., 2015a), onde cada objeto possa ser equipado com sensores e/ou atuadores para que possa ter sua própria identidade virtual e com capacidade potencial para comunicar, captar sinais, interagir e cooperar com o ambiente onde estão inserido de forma independente, interagindo com outros objetos para atingir um objetivo em comum (LOUREIRO GARRIDO, 2015). A conexão com a *internet* viabilizará gerir remotamente os objetos, permitindo que os objetos sejam acessados como provedores de serviços (SANTOS, 2016), contribuindo para concretização de novos domínios de aplicação das Tecnologias de Informação e Comunicação (TICs) (PIRES et al., 2015), tornando possível automatizar tarefas domésticas e melhorar a tomada de decisão nas aplicações a partir de uma melhor visão dos processos que ocorrem no ambiente onde o objeto de IoT está inserido (ANTUNES et al., 2016).

Um dos desafios mais importantes no gerenciamento da IoT será tratar a quantidade incalculável de objetos heterogêneos que geram uma grande e crescente quantidade de dados, segundo dados da *International Telecommunication Union*(ITU). Em 2020 o número de dispositivos conectados poderá chegar a 40 bilhões (SCHEUNEMANN, 2016). Uma solução para o contínuo crescimento da rede é o uso do protocolo IP na versão 6(IPv6). O IPv6 possui um número de endereçamento, que atenderá por muito tempo as necessidades da Internet. O IPv6 possui 128 bits no seu campo de endereço, o que acarreta 3,4x1038 IPs (SILVA et al., 2015a).

Além disso, esta massa de dispositivos ainda traz consigo diversos desafios acerca da heterogeneidade tanto em termos de *hardware* quanto de *software*, além dos diferentes protocolos de comunicação, formatos de dados, capacidade de memória e processamento, etc. Existe também fatores como a imprecisão dos dados produzidos (sistemas de *Radio-Frequency Identification* - Identificação por radiofrequência (RFID)) podem gerar entre 60 e 70% de dados incorretos), o grande conjunto de dados oriundos de redes de sensores produzidos em tempo real (ANTUNES et al., 2016: SILVA et al., 2015b).

Neste contexto o uso de *middleware* têm surgido como solução promissora para prover interoperabilidade e gerenciar a heterogeneidade de dispositivos IoT associados a aplicações (PIRES et al., 2015), visando abstrair as especificidades de tais dispositivos, promovendo a interoperabilidade entre eles, tornando o processamento contextual mais transparente para as aplicações e usuários final (SCHEUNEMANN, 2016). Entretanto, eles dependem fortemente de uma base de conhecimento que contenha informações sobre sensores, atuadores, fabricantes, conceitos físicos, unidades físicas, modelos de dados, modelos de erros, entre outros (PIRES et al., 2015).

A adoção de uma plataforma de *middleware* também pode contribuir para facilitar a construção de aplicações para IoT (PIRES et al., 2015). Tais plataformas são inseridas entre as aplicações e a infraestrutura (de comunicação, processamento e sensoriamento) subjacente, responsáveis por processar informações, tais como *status*, localização, funcionalidades e gerando uma grande quantidade de dados (*big data*), provendo um meio padronizado para o acesso aos dados armazenados em servidores (*WebService*) e serviços fornecidos pelos objetos através de uma interface de alto nível.

Existem várias opções de plataformas para IoT, entre elas podemos citar: *ManIoT, EcoDiF, S3OIA, Link Smart, CARRIOTS, AWS IoT, Evothings***,** entre outras. Estas plataformas possuem suas características e recursos personalizados, fornecendo serviços específicos e diferenciados para se adaptar as necessidades do órgão contratante.

1.1 Objetivo

Este trabalho propõe uma plataforma de hospedagem e gerenciamento de serviços de Internet das Coisas (IoT), que integrará e gerenciará os dispositivos de uma rede IoT, que permitirá a coleta e controle de informações de diferentes dispositivos, seguindo a premissa da “Internet das Coisas”.

1.2 Objetivos específicos

- Projeto e implementação de um módulo para cadastro e edição de usuários, projetos, recursos e *dashboard*;
- Armazenamento de dados vindos dos recursos/dispositivo físicos.
- Implementação de um módulo de coleta e armazenamento de dados vindos dos recursos/dispositivo físicos;
- Projeto e implementação de um módulo para visualização dos dados coletados para um projeto previamente cadastrado.

2. Revisão Bibliográfica

1.3 Internet das Coisas – IoT (*Internet of Things*)

1.3.1 História

Com o passar dos anos presenciamos direta e indiretamente evoluções tecnológicas, como por exemplo, a evolução do *hardware*, que obteve um impressionante progresso no campo de dispositivos embarcados. Objetos físicos tais como eletrodomésticos equipados com sensores sem fio e atuadores que podem se conectar a internet (DE FRANÇA et al., 2011), estão cada vez menores e possuem mais recursos e capacidade de processamento (SANTOS, 2017).

Além da evolução tecnológica do hardware, a internet tornou-se parte fundamental da sociedade, presente de forma ativa nas práticas cotidianas, permeando seus afazeres e mediando suas conexões com outros humanos, fazendo com que em um futuro próximo a internet e os dispositivos se tornem cada vez mais transparentes para os seus utilizadores (SILVA et al., 2015ª: SANTOS, 2014).

A Internet das Coisas, em poucas palavras, nada mais é que uma extensão da Internet atual, composta por um conjunto de objetos ”coisas”, capazes de comunicar, monitorizar e partilhar informação entre si (VALENTE, 2017), com o objetivo de integrar o estado dos objetos que constituem o nosso mundo em aplicações de software, beneficiando do contexto onde estão instaladas, permitindo a existência de uma rede totalmente conectada, gerando informação qualificada de situações, lugares e objetos (SANTOS, 2014). Como exemplos de possibilidade, a IoT permitirá, por exemplo, controlar remotamente os objetos e permitir que os próprios objetos sejam acessados como provedores de serviços (SANTOS, 2016), onde as informações das ”coisas” são analisadas e utilizadas para iniciar ações de forma a criar um sistema mais inteligente e autônomo (PEREIRA, 2017).

Com surgimento de novas aplicações que possibilitam o acesso e a interação com uma grande quantidade de dispositivos heterogêneos, a IoT permitirá o desenvolvimento de aplicações em diversos domínios, dentre eles temos a automação domiciliar e industrial, sensoriamento em diversos tipos de ambientes, aplicações médicas como coleta de dados de pacientes e monitoramento de idosos, controle de trânsito. Estas aplicações farão uso da grande quantidade de dados

gerados por estes dispositivos e que requerem a mínima intervenção do usuário para o seu funcionamento (SCHEUNEMANN, 2016). Espera-se, assim, que a solução obtida tire partido dos mais recentes protocolos de rede e que consiga ser aplicado em qualquer situação real de forma a melhor aperfeiçoar a solução (PEREIRA, 2017).

1.3.2 Evolução

A Internet das Coisas vem ganhando grande destaque no cenário das telecomunicações e está sendo considerada a revolução tecnológica que representa o futuro da computação e comunicação.

O Conselho Nacional de Inteligência dos EUA (NIC) a considera o IoT como uma das seis tecnologias civis com potenciais impactos nos interesses dos EUA até 2025. O NIC prevê que em 2025 todos os objetos do cotidiano poderão estar conectados à internet (SILVA et al., 2015a).

A Internet das Coisas trará diversas de possibilidades de avanços tecnológicos nas mais diversas áreas, tais como médica, automotiva, petrolífera e de serviços, entre outras, despertando interesses e ativando a imaginação para novas oportunidades (BALAGUER, 2014). Dentro deste cenário, várias pesquisas vêm sendo desenvolvidas e consolidando tendências, criando um amplo ambiente de promissores resultados para a transformação da informação em rede (SILVA et al., 2015a).

A Internet das Coisas terá alto impacto em vários aspectos da vida cotidiana e do comportamento dos potenciais usuários (BANDYOPADHYAY; SEN, 2011), a IoT está surgindo como uma nova forma de aplicações que auxiliarão os seres humanos nas tarefas do dia a dia (SILVA et al., 2015a). Os efeitos do IoT serão visíveis em ambos os campos de trabalho, domésticos e empresarias. No contexto domestico a vida assistida, casas inteligentes e escritórios, saúde, aprendizagem melhorada, etc. No contexto empresarial: na automatização e na produção industrial, na logística, na gestão dos processos de negócio, o transporte inteligente de pessoas e bens, etc (BANDYOPADHYAY; SEN, 2011).

No entanto, para que a visão da IoT se torne uma realidade, as seguintes questões precisam ser tratadas:

- Como garantir a interoperabilidade entre dispositivos;
- Como proporcionar inteligência aos dispositivos;
- Como permitir a adaptação e comportamento autômato;
- Como garantir confiança, segurança e a privacidade dos usuários e dos seus dados;
- Será feita a utilização eficiente dos recursos em dispositivos de baixa potência limitada.

1.3.3 Componentes básicos

Um objeto inteligente é um dispositivo equipado com uma unidade de sensor e/ou atuador, uma unidade de processamento, uma unidade de comunicação e uma fonte de energia. A unidade de sensor e/ou atuador dá ao objeto inteligente a capacidade de interagir com o mundo físico. A unidade de processamento permite que o objeto inteligente transforme os dados capturados pelos sensores. A unidade comunicação permite o objeto inteligente comunique suas leituras de sensores com o mundo exterior e receber entrada de outros dispositivos objetos inteligentes, através de *Radio-Frequency IDentification (*RFID). A fonte de energia fornece a energia elétrica para que o objeto inteligente faça seu trabalho (VASSEUR & DUNKELS, 2010: SANTOS, 2017).

1.3.4 Desafios

A possibilidade de novas aplicações IoT é crescente, mas temos novos desafios de conectar à Internet objetos heterogêneos com implementações diferentes e com restrições de processamento, memória, comunicação e energia. Desta maneira, é preciso fazer uma adaptação nos protocolos existentes para atender requisitos da IoT. Além disso, os objetos inteligentes não seguem os mesmos padrões de comunicação e roteamento como o da Internet (SANTOS, 2017).

Com o surgimento de novas aplicações para IoT, surgem também novos desafios, as aplicações deveram, ser capazes de lidar com os desafios em relação

as imperfeições, heterogeneidade e inconsistências nos dados obtidos dos objetos. Os desafios impostos por essas novas aplicações devem ser explorados e novas soluções devem ser propostas para que a IoT contemple as suas expectativas e todo seu potencial (SANTOS, 2017).

Os computadores atuais em sua grande maioria são dependentes da interação humana para obter e gerar informações, isto é um problema, pois as pessoas possuem limitação de tempo e podem gerar erros na inserção dos dados (PEREIRA, 2017). Como a chegada da IoT, os objetos inteligentes conseguiram obter informações mais precisas e confiáveis em tempo real sem a necessidade de interação humana. Com isso uma quantidade incalculável de objetos inteligentes irão gerar uma massiva e crescente quantidade de dados, que deverá ser processado pela IoT (SILVA et al., 2015b).

A heterogeneidade é um dos principais desafios a serem tratados por uma plataforma de mediação de dados. Em IoT, não há padronização entre os dispositivos, seja em relação aos protocolos de comunicação, ao modo de operação ou aos modelos de dados usados. Em especial, os dados coletados podem estar em diferentes formatos (texto, tabelas, objetos, etc) e apresentarem sintaxes e semânticas distintas. As plataformas de mediação devem permitir a integração e interoperabilidade dos diferentes modelos tecnológicos, semânticos e lógicos existentes (SILVA et al., 2015b).

1.3.5 Principal problema

A desvantagem relevante da IoT é no que diz respeito à questão de privacidade e segurança. Os objetos inteligentes têm a capacidade de utilizar muitos dados e informações sobre um usuário. Esses dados podem ser dados pessoais de grande importância. Se outras pessoas ou dispositivos tiverem acesso a esses dados, pode ocasionar grandes danos (BANAFA, 2015).

1.4 Ipv6

O *internet protocol*(IP) é a chave para o funcionamento da Internet e o principal responsável por ela fazer um uso muito mais inteligente das tecnologias

que o utilizam,ou seja o IP é o coração da Internet. O IP identifica cada dispositivo na rede, com um endereço numérico globalmente único (MOREIRAS, 2014). Atualmente a versão mais utilizada do protocolo é a versão 4, que possui 32 bits de endereçamento, o que possibilita, aproximadamente 4 bilhões de endereços de rede. Devido a grande e crescente quantidade de dispositivos conectados na internet, acarretou o esgotamento de endereços disponíveis (SILVA et al., 2015a).

Uma solução para o contínuo crescimento da rede é o uso do protocolo IP na versão 6 (IPv6), que possui 128 bits de endereçamento, o que possibilita 2128 (~341 decilhões) de endereços IPs disponíveis (MELO, 2012).O IPv6 atenderá por muito tempo as necessidades da Internet (SILVA et al., 2015a).

É fato que o IPv6 resolverá o problema de limitação do endereçamento IPv4, neste aspecto vale a pena destacar que o IPv6 gerará um novo paradigma que será conhecido como a era da “Internet das Coisas”, pois haverá disponibilidade para qualquer pessoa, empresa e quaisquer dispositivos que precisem ser conectados à Internet sem preocupação com a economia de IP (MELO, 2012).

1.5 Plataformas

Plataformas de hospedagem e gerenciamento de serviços para IoT tem como principal objetivo conectar e unificar inúmeros objetos e sistemas heterogêneos, permitindo a coleta e o processamento de informações em larga escala. Todos os dados coletados devem ser compartilhados para que aplicações e serviços, que não precisam necessariamente ter conhecimento ou domínio das tecnologias envolvidas, possam utilizá-los (SILVA et al., 2015b).

Devido ao grande volume de dispositivos e dados é importante que as plataformas utilizem mecanismos para garantir que os dados sejam coletados de forma adequada, integrando-os e armazenando-os, em alguma forma escalável de armazenamento com a ajuda de infraestruturas de *Cloud Computing* e bases de dados no SQL, permitindo a análise por diversos mecanismos que possam estabelecer correlações e gerar informação de maior valor agregado (LI et al., 2012).

As plataformas para IoT usualmente fornecem serviços de registro, monitoramento e descoberta para dispositivos, *gateways* e aplicações de controle.

Estes serviços permitem que dispositivos sejam capazes de publicar detalhes sobre seus serviços e ser encontrados por outros dispositivos ou usuários (ANTUNES et al., 2016).

Pode-se perceber que a Internet das Coisas é uma promessa que já tem apresentado alguns projetos importantes e que a tecnologia vem apresentando soluções que sustentará um enorme potencial de desenvolvimento (SILVA et al., 2015a).

1.5.1 Objetivos das plataformas de IoT

A heterogeneidade dos elementos de hardware e software da IoT torna o uso de *middlewares* essencial para prover a interoperabilidade entre os dispositivos e facilitar o desenvolvimento de aplicações (RAZZAQUE et al., 2016). Um *middleware* é uma camada de *software* utilizada entre aplicações e os provedores de informações e serviços baseados em tecnologias de internet. O mesmo agrupa funções comuns entre aplicações, em áreas como: interoperabilidade, gerenciamento de dispositivos, segurança e ciência de situação (SCHEUNEMANN, 2016).

Devido ao grande volume de dispositivos e dados é importante que as plataformas utilizem mecanismos para garantir que os dados sejam coletados de forma adequada, integrando os e armazenando-os, em alguma forma escalável de armazenamento com a ajuda de, por exemplo, infraestruturas de *Cloud Computing* e bases de dados, permitindo a análise por diversos mecanismos (mineração de dados, contexto computacional) que possam estabelecer correlações e gerar informação de maior valor agregado (SILVA et al., 2015b).

Plataformas de mediação de dados podem ser utilizadas para facilitar o desenvolvimento de aplicações e a integração das diferentes tecnologias no âmbito da IoT. Entre as funcionalidades fornecidas, há o provimento de serviços para gerenciar diversos dispositivos, auxiliar no armazenamento e na recuperação de dados, e processar dados usados para disparar cálculos, estatísticas, alertas e muito mais.

1.5.2 Exemplos de plataformas para Internet das Coisas

1.5.2.1 ManIoT (*Management for Internet of Things*)

A plataforma ManIoT integra e gerencia as funcionalidades individuais dos dispositivos em uma rede Internet das Coisas e permite a criação de novos serviços cientes do contexto.

A ManIoT define uma estrutura de gerenciamento em dois escopos: gerenciamento local, onde gerente local atua dentro de um cenário, gerenciando os dispositivos que compõem este cenário e gerenciamento global, onde o gerente global/remoto procura uniformizar as ações realizadas em diferentes cenários a partir de diretivas de alto nível, controlando os dispositivos remotamente.

Além disso, a plataforma prevê serviços genéricos, tais como descoberta de nós, armazenamento de dados e autenticação, que são blocos básicos para a construção de aplicações Internet das Coisas. Ainda, visando a extensibilidade e integração com outros sistemas, a plataforma faz uso de protocolos e padrões populares da indústria para modelos de dados, como o XML (*eXtensible Markup Language*) e o REST (*Representational State Transfer*)(ANTUNES et al., 2016).

A plataforma ManIot também leva em conta a heterogeneidade dos dispositivos . Assim, ManIoT não requer grandes modificações ou instalações de softwares adicionais na maioria dos dispositivos da rede ou nos dispositivos dos usuários. O acesso ás aplicações da plataforma ManIot é realizado através de uma interface Web.

1.5.2.2 EcoDiF - *Web Ecosystem of Physical Devices*

O EcoDiF (Ecossistema Web de Dispositivos Físicos) é uma plataforma baseada na Web para integrar dispositivos físicos heterogêneos com aplicativos e usuários.

Os principais objetivos do EcoDiF são integrar dispositivos físicos heterogêneos e fornecer funcionalidades como controle de dados em tempo real, visualização, processamento e armazenamento, permitindo seu uso em vários domínios de aplicação.

O EcoDiF está alinhado com o paradigma *Web of Things* (WoT), que utiliza tecnologias e protocolos da Web existentes para permitir a inclusão de dispositivos físicos no mundo digital, para que seus dados e serviços sejam reutilizados em diferentes aplicativos como qualquer recurso da Web.

O EcoDiF foi projetado e implementado com base nos princípios REST e depende de padrões e protocolos atuais da Web, como HTTP e URIs. A adoção da abordagem REST permite que o EcoDiF padronize e simplifique os processos de desenvolvimento de aplicativos. Os dados fornecidos por dispositivos são estruturados usando a linguagem EEML (*Extended Environments Markup Language*), que fornece um formato simples baseado em XML para descrever dados obtidos de dispositivos em um contexto específico.

A arquitetura de software da EcoDiF é composta por sete módulos:

- Modulo de colaboração;
- Módulo de conexão de dispositivos;
- Modulo de aplicação;
- Modulo de armazenamento;
- Modulo de visualização e gerenciamento;
- Modulo de manipulação de dados;
- Modulo de serviços comuns.

1.5.2.3 S3OIA - *Smart Spaces and Smart Objects Interoperability Architecture*

O S3OIA é apresentado como uma arquitetura sintática / semântica orientada a serviços que permite a integração de qualquer tipo de objeto ou dispositivo da Internet das coisas. Essa arquitetura possibilita o uso de recursos heterogêneos subjacentes como substrato para a composição automática de aplicações complexas através de um paradigma semântico do espaço triplo.

Os aplicativos criados são dinâmicos e adaptáveis, pois podem evoluir dependendo do contexto em que são executados. O cenário de validação desta

arquitetura engloba áreas que são propensas a envolver seres humanos para promover a autonomia pessoal (VEGA-BARBAS et al., 2012).

Devido a capacidade limitada de alguns dispositivos IoT, esta arquitetura provê a utilização de *gateways*, que podem realizar funções de controle e gestão de recursos para esses dispositivos, além de realizar a abstração destes e normalização dos dados, de forma a tratar a heterogeneidade (DAVET, 2015).

A arquitetura do S3OIA está organizada em módulos e possui cinco grupos funcionais (VEGA-BARBAS et al., 2012):

- Descoberta de dispositivos e serviços
- Exposição Semântica dos Espaços Triplos e Serviços Web
- Reposição de serviços e resolução de dependências
- interface de interação
- Composição, Tolerância a Falhas e Dependências Distantes

1.5.2.4 Link Smart

O LinkSmart é uma estrutura e uma infraestrutura de serviços para a criação de aplicativos distribuídos da Internet de Coisas, é um *middleware* desenvolvido no âmbito do projeto Hydra para sistemas integrados em rede, permitindo que os desenvolvedores incorporem dispositivos físicos heterogêneos em seus aplicativos através de serviços web fáceis de usar para controlar qualquer dispositivo (LANG, 2014).

Sua arquitetura possui três camadas principais: (i) camada de rede: responsável pela comunicação com os dispositivos; (ii) camada de serviço: responsável pelo gerenciamento de eventos, dispositivos, escalonamento de recursos, dentre outros; (iii) camada semântica.

A arquitetura do LinkSmart é representada pelos componentes: *Gateway, Network manager, Event manager, Crypto and trust, Device application catalogue(DAVET, 2015).*

1.5.2.5 CARRIOTS

A CARRIOTS (PIRES, 2015) é uma plataforma para aplicações em Internet das Coisas que utiliza serviços de nuvem para gerenciar dados providos por qualquer tipo de dispositivo, tendo como objetivo coletar e armazenar qualquer dado originado dos mais diversos dispositivos utilizá-lo em seu motor de aplicações e disponibilizá-lo a seus usuários não importando o volume de dispositivos conectados.

A CARRIOTS conecta dispositivos a dispositivos e a outros sistemas, o que a faz se alinhar ao conceito de *Platform as a Service (PaaS)* da computação em nuvem. Portanto, se um sistema for conectado à plataforma, ele também pode ser modelado como um dispositivo (DAVET, 2015).

A plataforma CARRIOTS foi projetada tendo como base dois blocos principais:

- (i) organização lógica de entidades;
- (ii) arquitetura de componentes de software.

A arquitetura de software do CARRIOTS é formada pelos seguintes módulos:

- API REST;
- Big Data;
- Gerenciamento de projetos e dispositivos;
- Processamento de eventos e e regras de negocio;
- Segurança;
- *logs* e *debug*;
- Painel de controle;
- Módulo de comunicação externa.

1.5.3 Comparação entre as plataformas existentes

Existem várias propostas de plataformas para IoT, cada uma atendendo um subconjunto dos requisitos necessários para viabilizar a comunicação entre dispositivos heterogêneos. O desenvolvimento de plataformas de middleware especificamente voltadas para ambientes de IoT é uma área de pesquisa recente que tem atraído a atenção da indústria e da comunidade acadêmica (PIRES, 2008).

No contexto de IoT, a falta de padronização existente na área faz com que tais plataformas de middleware adotem diferentes modelos de programação que, em geral, não são compatíveis entre si (ANTUNES, 2016). Outras limitações das soluções existentes dizem respeito ao fato de elas não abordarem requisitos de escalabilidade de forma apropriada, fornecerem modelos inadequados de governança, e negligenciarem questões de privacidade e segurança na sua concepção. Finalmente, faz-se ainda necessária a inclusão de mecanismos para lidar com a massiva quantidade de dados (também potencialmente heterogêneos) produzidos pelos dispositivos interconectados (DELICATO, 2013).

1.5.3.1 Requisitos de Middleware para IoT

No contexto de IoT, plataformas de middleware devem satisfazer a um conjunto de requisitos e desafios emergentes neste cenário, visando atender às necessidades de aplicações e usuários. Esta seção descrever alguns requisitos relevantes para plataformas de middleware em IoT (BANDYOPADHYAY et al. 2011)

I. <u>Interoperabilidade;</u>

A interoperatividade é um dos desafios para a concretização do paradigma de IoT devido ao grande número de dispositivos a serem integrados e sua heterogeneidade tanto em termos de hardware quanto de software, protocolos (muitos deles proprietários), esforço em reconciliar formatos de dados dentro e através dos sistemas (PIRES, 2008). Caso haja uma rede fragmentada, com restrições de interoperabilidade entre os dispositivos, teremos uma limitação no potencial da IoT, dessa forma é preciso um determinado nível de interoperabilidade para ter um ecossistema de IoT bem-sucedido, fornecendo um intercâmbio e/ou fusão de dados, desempenhando um papel importância no cenário de IoT por

permitindo aplicações com maior valor agregado para os usuários, fazendo uso de diversos dispositivos e/ou outras aplicações. (SANTOS, 2016: DELICATO, 2013).

II. Descoberta e gerenciamento de dispositivos;

É importante que uma plataforma de middleware possibilite a descoberta de dispositivos presentes no ambiente em questão, realizada dinamicamente a fim de atender os requisitos das aplicações. Além disso, é necessário prover mecanismos para o gerenciamento de dispositivos, que diz respeito à capacidade de fornecer informações de localização e estado do dispositivo permitindo, dentre outras funcionalidades, desconectar algum dispositivo roubado ou não reconhecido, atualizar software embarcado, modificar configurações de segurança, modificar remotamente configurações de hardware, localizar um dispositivo perdido, apagar dados sensíveis de dispositivos, e até mesmo possibilitar a interação entre dispositivos.

III. Ciência de contexto;

A ciência do contexto é a coleta de entradas capazes de caracterizar um usuário, estado de um objeto e seus vizinhos, refletindo as condições atuais do ambiente no qual o mesmo se encontra e do próprio dispositivo computacional utilizado (ANTUNES, 2016). Plataformas de middleware em IoT são responsáveis pela coleta, gerenciamento e processamento de informações de contexto providas por múltiplas fontes, liberando as aplicações e usuários da tarefa de manipulá-las.

IV. Escalabilidade;

A Escalabilidade se refere à capacidade que um ambiente possui de entregar volumes em transformação de cargas de trabalho ao ajustar os recursos do sistema, de modo que seja capaz de assimilar um número crescente de dispositivos e requisições e funcionar corretamente, mesmo em situações de uso intenso (TOST, 2013). Permitindo que ela possa acompanhar a demanda de coleta e análise de dados e, consequentemente, prover respostas, decisões e/ou atuações de maneira eficiente.

V. Segurança;

O requisito de segurança é subdivido em três grupos de desejáveis de segurança: Confidencialidade, Integridade, Disponibilidade (ANTUNES, 2016). A fim de proteger tanto os dispositivos envolvidos quanto os recursos expostos à rede (SANTOS, 2016).

- Confidencialidade – requisito onde os dados transmitidos podem ser "escutados" e entendidos por elementos participantes da comunicação, onde elementos sem autorização não tem acesso aos dados transmitidos.
- Integridade – os dados não podem ser alterados por elementos da rede sem devida autorização. Implementa-se integridade criptografando as mensagens e verificando-as no lado do receptor.
- Disponibilidade – deseja-se manter o sistema sempre disponível.

Na Tabela 1 símbolo "V" denota que o requisito é completamente atendido, o símbolo "O" indica que o requisito é parcialmente atendido, e o símbolo "X" indica que o requisito não é atendido.

Tabela 1 - Tabela de Requisitos versus Plataformas IoT

Plataformas	Interoperabilidade	Descoberta e Gerenciamento de Dispositivos	Ciência de Contexto	Escalabilidade	Segurança
ManIoT	V	O	V	V	V
EcoDiF	V	O	O	V	V
Carriots	V	V	O	V	V
LinkSmart	V	V	V	X	V
S_3OIA	V	V	V	X	X
Academic-IoT	V	O	O	X	X

Diante do exposto na Tabela 1, conclui-se que nenhuma plataforma de IoT foi capaz de atender a todos os requisitos levantados, tratando apenas subconjuntos dos requisitos, abordando-os de formas diversas.

A interoperabilidade foi um requisito atendido por todas as plataformas avaliadas, sendo um requisito fundamental para a concretização do paradigma IoT.

A ciência de contexto é abordada pela maioria das plataformas, mesmo que de maneira parcial, caracterizado pelos dados coletados através da inclusão de

dados semânticos, como localização, horário de coleta e afins. Possibilitando o envio de informações de descoberta e gerenciamento de dispositivos da aplicação.

A escalabilidade foi um requisito foi pouca atendida devido à dificuldade de coleta e processamento da massiva quantidade de dados enviados pelos dispositivos.

Foi possível concluir que o estado da arte para plataformas de middleware para IoT ainda se encontra em uma fase inicial, onde as tecnologias e abordagens ainda divergem, além de não haver uma solução capaz de englobar todos os requisitos necessários para a concretização do paradigma, no qual requisitos importantes ainda não foram exaustivamente explorados, como a segurança em plataforma IoT. Existem várias pesquisas e desenvolvimentos em aberto.

1.6 Protocolos para Internet das Coisas

Os protocolos de comunicação são parte importante na implantação de infraestruturas Internet das Coisas, podendo contribuir para uma melhor utilização de recursos como os de processamento, memoria, energia e banda de comunicação dos dispositivos inteligentes, tendo em vista fatores inerentes do universo *Internet* das Coisas como heterogeneidade, mobilidade, intermitência de comunicação, extensibilidade, escalabilidade e capacidade de autoconfiguração (DAVET, 2015).

1.6.1 CoAP

O CoAP (*Constrained Application Protocol*) é um protocolo de transferência de dados da web especializado para uso com nós restritos e redes restritas, baseia-se na arquitetura REST, com o objetivo de se ter um protocolo web genérico, alternativo ao HTTP, adequado aos requisitos especiais do ambiente introduzido pela IoT e caracterizado por dispositivos de baixa potência, considerando principalmente, a questão do consumo de energia (SHELBY; HARTKE; BORMANN, 2017).

Um dos principais objetivos do CoAP é projetar um protocolo web genérico para os requisitos especiais de ambiente com restrições, principalmente em relação

a energia, automação de construção e outros aplicativos de máquina a máquina (M2M).

Principais Características do CoAP :

- Protocolo da Web que cumpre os requisitos M2M em ambientes restritos
- UDP vinculativo com confiabilidade opcional que admite solicitações *unicast* e *multicast.*
- Trocas de mensagens assíncronas.
- O URI e o suporte ao tipo de conteúdo.
- O Recursos simples de *proxy* e *cache*.
- Mantém a mensagem pequena, limitando assim a necessidade de fragmentação.
- Define quatro tipos de mensagens: Confirmar, Não confirmar, Reconhecer, Redefinir(SHELBY; HARTKE; BORMANN, 2017).

1.6.2 UPnP

O UPnP (Universal Plug and Play) usa uma abordagem que prove funcionalidades que minimizam os esforços de gerencia dos dispositivos ou objetos inteligentes conectados, como a ativação, desativação e descoberta destes dispositivos e seus recursos por parte das aplicações(DAVET, 2015).

As características de autoconfiguração e de descoberta automática de dispositivos inerentes ao UPnP, pois utiliza protocolos e padrões consolidados da Internet, tais como HTTP, SOAP e XML, que é independente de plataforma e de linguagem de programação, o que aumenta bastante a interoperabilidade entre dispositivos, tornando um facilitador na gerência do crescente número de dispositivos conectados, o que traduz o fato deste ser bastante utilizado no cenário da Internet das Coisas (VITERBO, 2007).

Componentes básicos do UPnP:

- Dispositivo UPnP;
- Serviço UPnP;

- Ponto de Controle UPnP.

Caso um dispositivo não suportar o padrão UPnP, um *gateway* UPnP pode ser usado para mapear os protocolos definidos pelo UPnP nos protocolos entendidos nativamente pelo dispositivo (DAVET, 2015). Também permite o controle de dispositivos e o processamento de serviços de forma descentralizada, e permite a interoperabilidade entre pontos de controle.

Protocolos específicos UPnP:

- TCP/IP;
- HTTP, HTTPU, HTTPMU;
- SSDP - *Simple Service Discovery Protocol;*
- GENA - *Generic Event Notification Architecture*;
- SOAP - *Simple Object Access Protocol*;

1.6.3 MQTT

MQTT (*Message Queuing Telemetry Transport)* é um protocolo de mensagens extremamente simples e leve, baseado em dois fatores chave: arquitetura *publish-subscribe* e tópicos. Utiliza o paradigma *publish/subscribe* para a troca de mensagens, no qual um cliente publica uma mensagem em um tópico e todos os outros clientes que são subscritos nesse tópico irão receber a mensagem.

O MQTT foi projetado para dispositivos embarcados com recursos limitados de memória e processamento e redes de baixa largura de banda, de alta latência ou não confiáveis. O MQTT utiliza o TCP/IP como protocolo de transporte, é independente do protocolo de aplicação utilizado e possui mecanismos de notificação de desconexão entre clientes [5].

Para usar o protocolo MQTT é necessário o uso de um *Broker*. Um *Broker* é um padrão de arquitetura que é responsável por coordenar a comunicação na rede, encaminhando requisições e mensagens para os clientes inscritos (ZYRIANOFF et al. 2017).

Um cliente publicar (*publish*) uma informação em um tópico. Quem tiver inscrito no determinado tópico (*subscribed*), irá receber a informação. Uma informação também pode ser publicada para mais de um tópico (MQTT, 2017).

Em seu desenvolvimento foram considerados alguns princípios, como o de minimizar requerimentos de recursos do dispositivo, bem como o de largura de banda tentando assegurar confiabilidade e garantia de entrega (DAVET, 2015). Esses princípios também tornam o protocolo ideal do mundo emergente "máquina-a-máquina" (M2M) ou "Internet das coisas" dos dispositivos conectados e para aplicações móveis onde a largura de banda e a energia da bateria são excelentes (MQTT, 2017).

Características do MQTT:

- Arquitetura *Publish/Subscribe;*
- Desacoplamento de Espaço;
- Desacoplamento de Tempo;
- Segurança;
- Níveis de *Quality of Service (QoS);*
- Aviso de Desconexão;
- Tópico de Subscrição Flexível;
- MQTT-SN;

3. Metodologia

A metodologia adotada para o desenvolvimento da plataforma foi o modelo evolutivo, pois o mesmo descreve o processo de desenvolvimento de modo que possa evoluir a partir de um protótipo inicial, onde são introduzidos aprimoramentos constantes na plataforma, obtendo assim uma avaliação continua dos impactos da inclusão de novas tecnologias poderão trazer ao serem implantadas na plataforma.

A plataforma foi projetada para ser executada em ambiente web, devido à facilidade no acesso, permitindo a vários usuários simultâneos o acesso às informações a partir de qualquer equipamento conectado à internet.

Para a modelagem da plataforma foi utilizada a Linguagem de Modelagem Unificada (*Unified Modeling Language* - UML) na qual exprime uma noção abstrata do comportamento da plataforma na forma de diagramas padronizados, definindo assim a estrutura da plataforma, onde foi adotado o diagrama de caso de uso, a fim de mostrar as funcionalidades do sistema. Para a modelagem do banco de dados foi utilizado o modelo de entidade e relacionamento (MER), para na qual fornece uma representação abstrata do banco de dados.

Para a construção da plataforma optou-se pelo uso do ambiente de desenvolvimento integrado NetBeans IDE 8.2. O ambiente NetBeans IDE 8.2 além de possuir um amplo numero de recursos avançados integra-se também a um conjunto de *framework's* interessantes para o desenvolvimento da plataforma, entre eles o Spring Web MVC, Eclipselink-JPA 2.1, JSTL-1.2.1, Apache Tomcat.

Para servidor optou-se pelo uso do Apache TomCat 8.0.27.0, por permitir a execução de aplicações web na linguagem de programação Java com especificação na tecnologia *Servlets* (extensão genérica do servidor, implementada como uma classe Java).

Para auxiliar no desenvolvimento da plataforma foi utilizado o *framework* Spring Web MVC 4.0.1, um *framework* completo para o desenvolvimento de aplicações Web. O Spring Web MVC utiliza o padrão de arquitetura MVC (*Model, Controller* e *View*), o que possibilita a divisão do projeto em camadas.

A camada *controller* é projetada ao redor do *DispatcherServlet*, cuja função é distribui as requisições necessárias para o tratamento das classes configuradas,

sendo responsável pelas consultas, mapeamentos e envios das solicitações aos *controller's* apropriados. Essas solicitações foram feitas com o uso formato JSON (*JavaScript Object Notation*) para as trocas de mensagens, onde algumas requisições foram feitas pelas *view's* com o uso da tecnologia AJAX, afim de prover trocas de mensagens de maneira assíncrona.

A camada *model* do Spring Web MCV foi integrada com o serviço de persistência de dados EclipseLink 2.7.1, para fazer acesso aos dados e realizar o mapeamento objeto-relacional da plataforma, configurado com o Sistema Gerenciador de Banco de Dados (SGDB) MySQL-8.0 com o uso da API Java Database Connectivity (JDBC) e do MySQL JDBC Driver.

A camada *view* do Sping Web MVC foi desenvolvida com a utilização de JSP (*JavaServer Pages*) integrado com JSTL-1.2.1(Biblioteca Padrão de Tags JSP), para a criação de paginas web. Foi ainda utilizado o mecanismos de folha de estilo CSS para modelagem da pagina e as bibliotecas javascript jQuery e Highcharts para manipulação das paginas, eventos e a construção de gráficos.

A comunicação entre os recursos e a plataforma foi implementada utilizando o protocolo MQTT, por ser relativamente simples usá-lo e possui comunicação assíncrona, desacoplando a plataforma do dispositivo. O MQTT possui características desejáveis como segurança, qualidade de serviço e facilidade de implementação, visto que a comunidade tem adotado esse protocolo para o âmbito IoT.

Foi utilizado o Moquitto (MOSQUITTO, 2018) como broker MQTT com ip 127.0.0.1 e porta 1883 executado no servidor local. O mosquito é responsável por receber todas as mensagens dos recursos, onde esses recursos podem ser um sensor de IoT em determinado ambiente ou um aplicativo em um data center que processa dados de IoT,.

Como cliente MQTT da plataforma foi utilizada a Biblioteca paho (PAHO, 2018), responsável por subscrever/assinar o tópico: "/AcademicIoT/Recursos/#", no qual receberá todas mensagens publicadas/enviadas pelos clientes conectados ao mosquito, o mosquitto encaminha as mensagens para o cliente Paho,

Foi adotado o envio de mensagens no formato JSON, disposto a padronizar o envio e recebimento de mensagens, facilitando a coleta dos dados vindos dos recursos clientes.

O Paho recebe as mensagens padronizada {dispositivo="dispositivo", tipo="tipo" , valor="valor", date="date”}, na qual é convertida e armazenada no banco de dados.

4. Resultados

A apresentação dos resultados foi organizada em duas seções. Na primeira são apresentados os resultados da modelagem do sistema, compreendendo o diagrama de casos de uso, as descrições dos casos de uso, diagrama entidade relacionamento e diagrama de sequência. Na segunda seção são apresentadas as telas de interface do sistema.

4.1 Diagrama de caso de uso

O diagrama de caso de uso é um modelo para uma representação das funcionalidades observáveis da plataforma do ponto de vista do usuário, sem ser necessário especificar como estas funcionalidades serão implementadas.

A ferramenta Astah Professional foi utilizada para modelagem do diagrama de caso de uso apresentado no Diagrama 1.

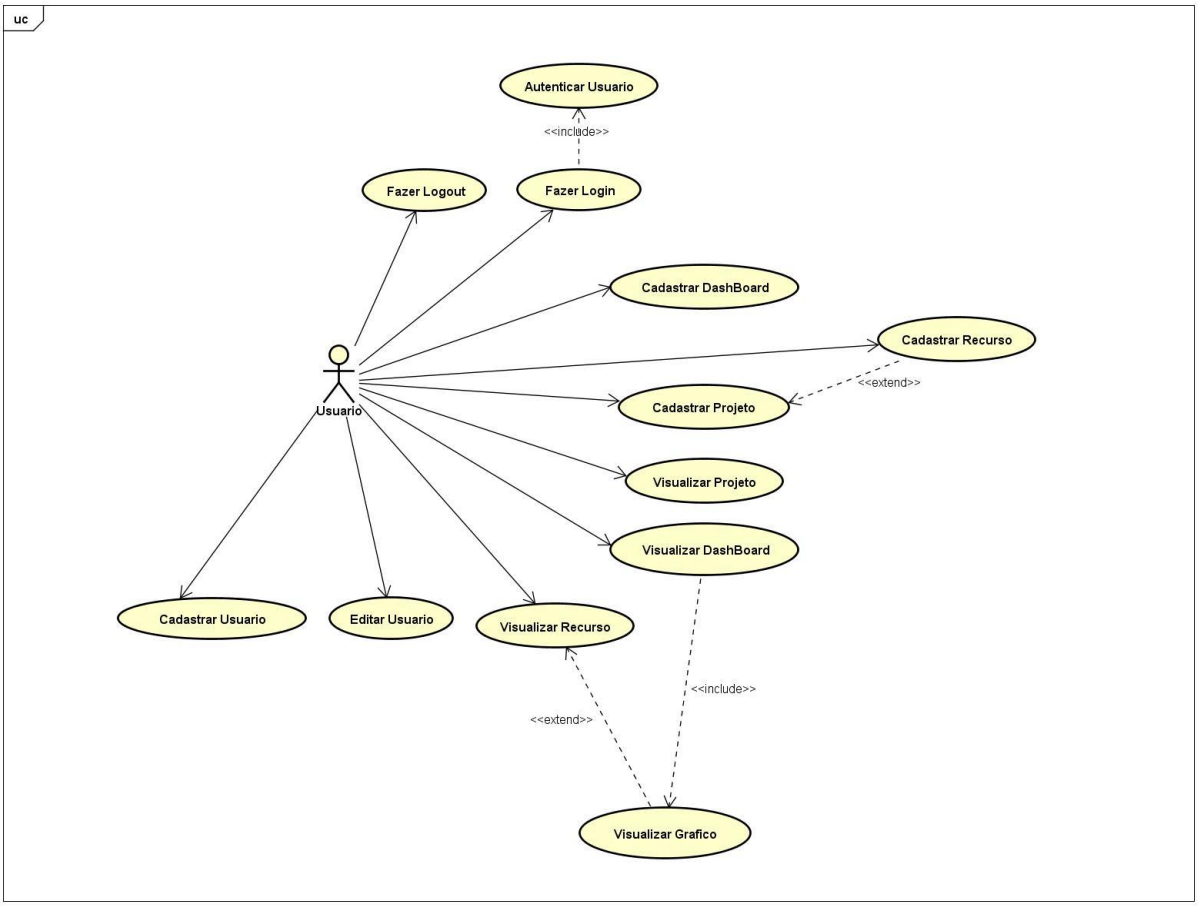

Diagrama 1 - Diagrama de Caso de Uso

UC01 - Fazer *Login*

Por meio da funcionalidade de fazer *login*, o usuário tem acesso a plataforma a partir da identificação do usuário com informações relativas à *login* e senha, com intuito que o usuário seja autenticado, a fim de que mesmo tenha acesso a funcionalidades da plataforma.

UC02 - Cadastrar Usuário

Por intermédio do cadastro de usuários, é possível fazer o armazenamento de informações dos usuários da plataforma, para que o mesmo tenha acesso à plataforma através a autenticação, para que possa utilizar as funcionalidades que a plataforma oferece.

UC03 - Editar Usuário

Através da funcionalidade editar usuário, o usuário pode alterar suas informações cadastradas, dados pessoais e de endereço.

UC04 - Cadastrar Recurso

Por meio do cadastro de recurso, o usuário poderá cadastrar informações descritivas sobre o recurso, com intuito de gerir os recursos, para que o mesmo possa ser armazenado em um banco de dados, para receber informações vindas de recursos físicos e/ou virtuais.

UC05 - Visualizar Recurso

Por intermédio da funcionalidade visualizar recurso, o usuário será capaz de visualiza dados coletados dos recurso cadastrados na plataforma na forma de um gráfico.

UC06 - Cadastrar Projeto

O cadastro de projeto permite que o usuário selecione e/ou cadastre um conjunto de recursos, com intuito de gerir e visualizar as informações relacionadas a este grupo de recurso, organizando em forma de um projeto, com possibilidade de adicionar ou remover um recurso do projeto.

UC07 - Visualizar Projeto

Com a visualização de projeto, o usuário pode selecionar um de seu(s) projeto(s), com intenção de visualizar informações dos recursos cadastrado, podendo o mesmo gerenciar recursos do projeto.

UC08 - Cadastrar *Dashboard*

O cadastro de *dashboard* propicia que o usuário selecione um conjunto de recursos vinculados a algum projeto, para a visualização de informações relacionadas a este grupo de recurso de maneira centralizada, deixando a cargo do usuário o agrupamento dos recursos, informando também uma descrição sobre o dashboard.

UC09 - Visualizar *Dashboard*

Através da funcionalidade visualizar *dashboard*, o usuário é capaz de selecionar um de seu(s) *dashboard('s)*, com intenção de visualizar informações de um conjunto de recursos cadastrado.

4.2 Diagrama Entidade Relacionamento (MER)

O modelo entidade relacionamento (MER) é a representação logica de como o banco de dados está disposto para um sistema. Para a geração do Diagrama 2, foi utilizado o MySQL Workbench, uma ferramenta para criação e administração de banco de dados MySQL.

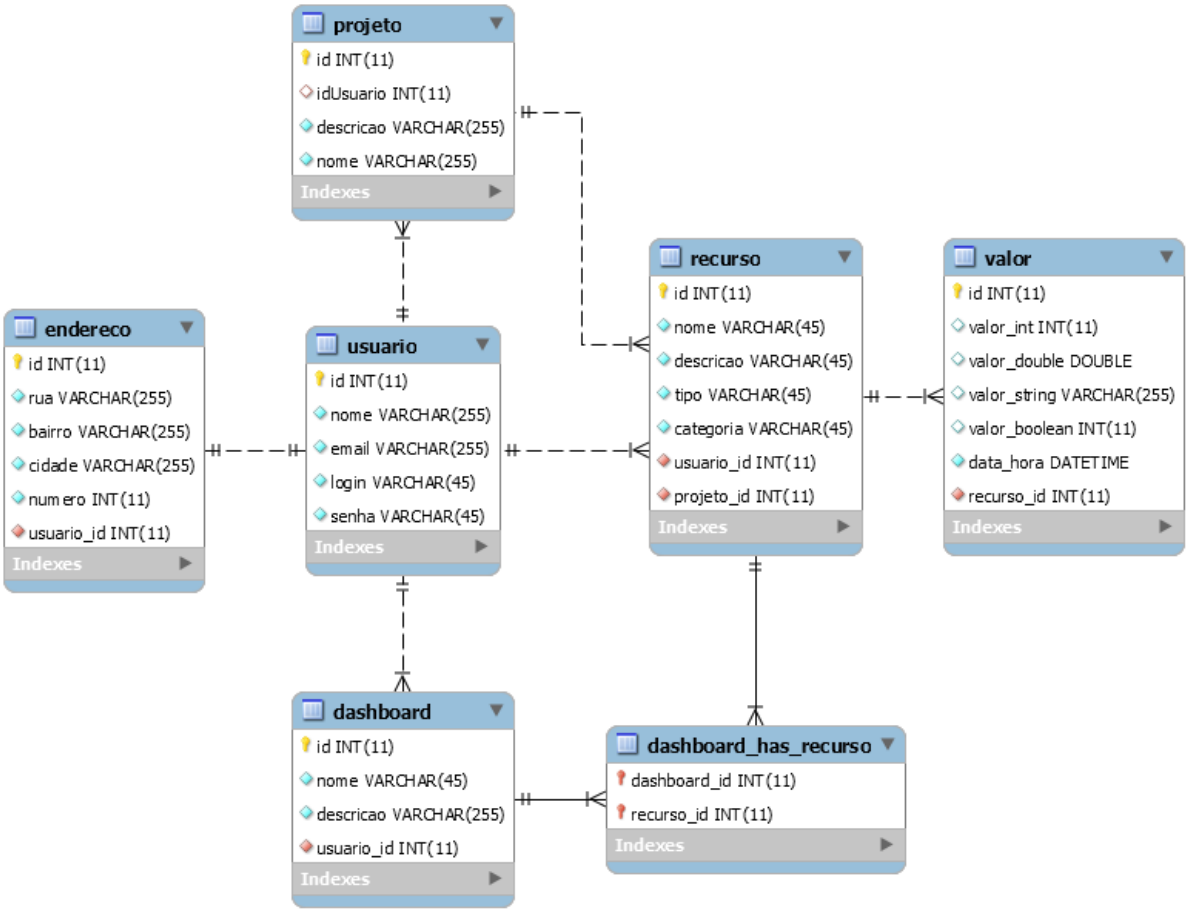

Diagrama 2 - Modelo Entidade Relacionamento

O Diagrama 2 de fornece uma representação visual do banco de dados, onde:

A tabela recurso é responsável por armazenar informações coletadas dos recursos dos usuários, armazenando essas informações na tabela variável. A tabela recurso mantem vinculo com as tabelas projeto e *dashboard*, armazenando a referência do projeto e do *dashboard* no qual o recurso pertencente.

A tabela projeto é responsável por armazenar informações relativas aos projetos cadastrados pelo usuário, permitindo que o projeto seja vinculado a um grupo de recursos do usuário.

A tabela usuário é responsável por armazenar informações referentes ao usuário, como informações pessoais e informação sobre endereço. A tabela usuário mantem vinculo com as tabelas projeto, recurso e *dashboard*, armazenando as referências de recursos, projetos e *dashboard* pertencentes ao usuário.

Projeto

Um projeto é uma entidade com intuito de organiza, gerir e visualizar as informações relacionadas a um grupo de recurso, com possibilidade de adicionar ou remover um recurso do projeto, bem como alterar as descrições do projeto.

Dashboard

Um *dashboard* é uma entidade que agrupa um conjunto de recursos em comum (ou não) de acordo com a necessidade do usuário, possibilitando o observar informações desse conjunto de recursos. Tem como objetivo agrupar/centralizar recursos, facilitando a visualização de um conjunto específico de recursos vinculados a algum projeto cadastrado pelo usuário.

Recurso

Um recurso é a representação de sensor, ou seja, um dispositivo que faz a detecção de variáveis provenientes de um ambiente físico, como luz, calor, movimento, umidade, pressão, etc. Estas variáveis são armazenadas em uma tabela do banco, onde fica contida data e hora de coleta dos dados, e o valor desse dados

(inteiro, real, string, booleano), onde está informações são entregues pelo protocolo MQTT.

4.3 Diagrama de Sequencia

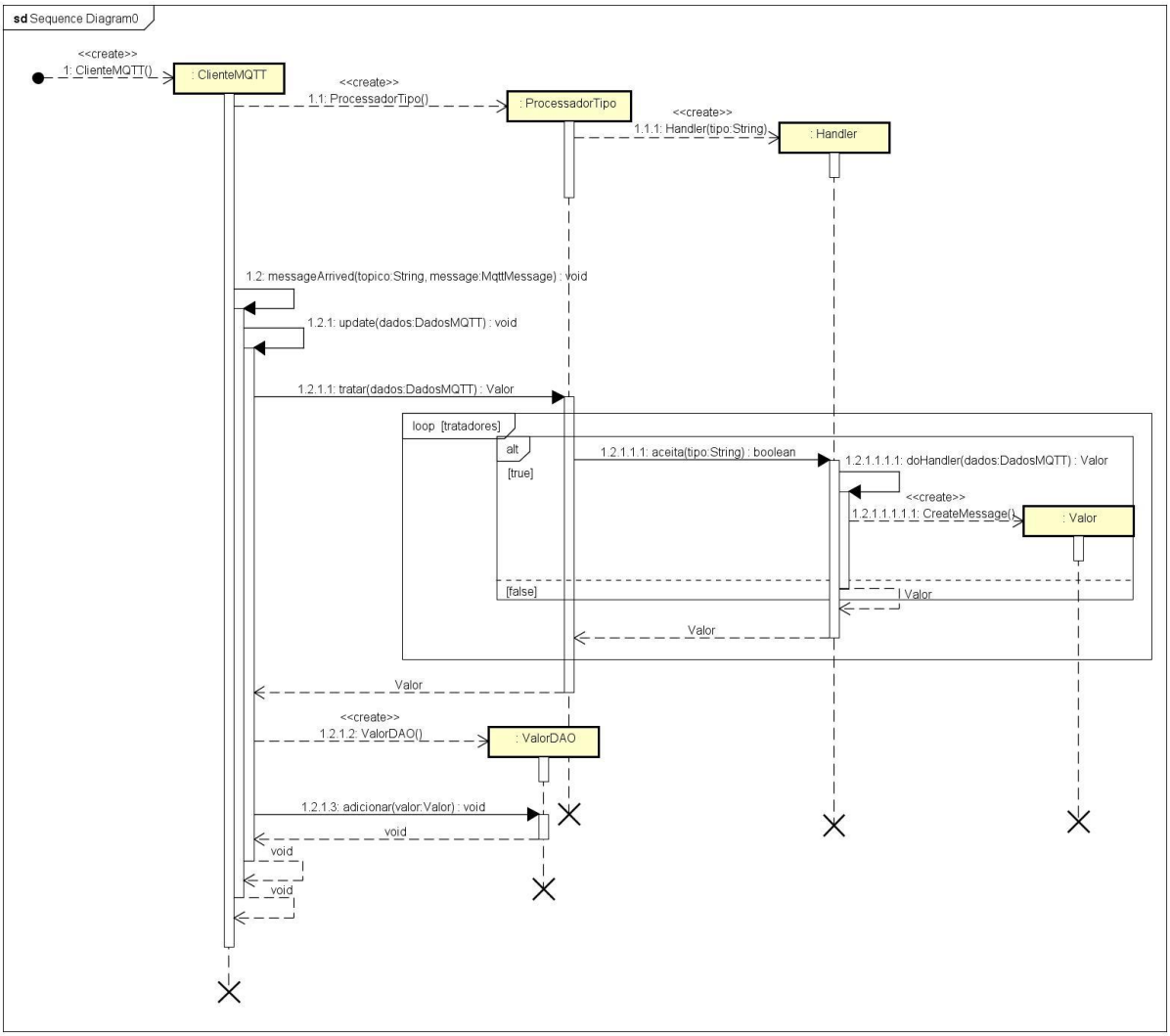

Diagrama 3 - Diagrama de sequencia

O Diagrama 3, descreve a sequencia de troca de mensagem após o recebimento de uma mensagem que contem o tipo do dados, o valor, identificador do recurso e a data/hora do envio, no qual esses valores são analisados, verificando o tipo do valor enviado e armazenando de maneira adequada no banco de dados.

4.4 Telas

4.4.1 Tela Inicial

A tela inicial mostrada na Figura 1 apresenta ao usuário as opções iniciais disponíveis, fica a disposição do administrador a parte da central da tela para exibição de noticias. Nesta tela persiste uma barra superior que é fixa, onde é disposto dois campos: “Inscrever-se” e “Entrar”. O campo “inscrever-se” ao ser selecionado, direciona o usuário para tela de cadastro de usuário mostrado na Figura 4, o campo “Entrar”, direciona a tela de *login* como mostra a Figura 2.

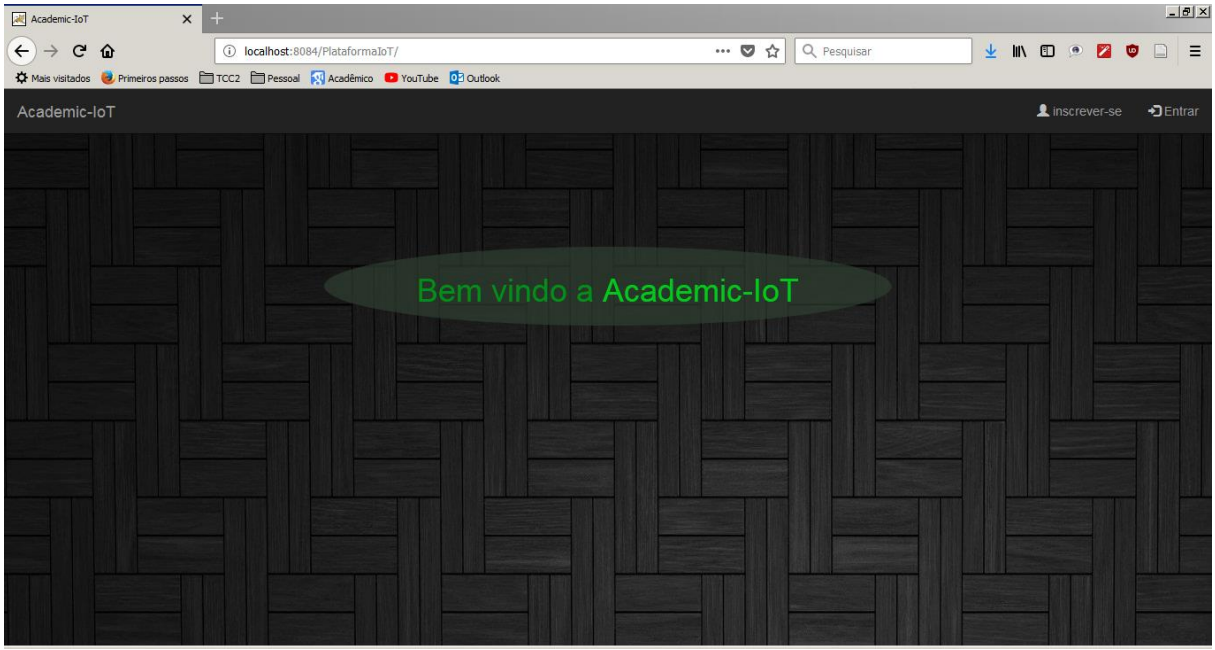

Figura 1 - Tela inicial

4.4.2 Tela Login

Antes de ser liberada a utilização das funcionalidades da plataforma, o usuário deverá se autenticar informando seu *login* e senha, para que seja validado, permitindo ou vedando o acesso, essa autenticação é feita através da tela de *login*, como mostra a Figura 2. Caso o usuário tenha esquecido a senha, é possível recuperar acessando o campo “esqueci minha senha”, na qual redireciona para tela de recuperação de senha, onde o usuário informa seu e-mail e em seguida clicar no campo “enviar”, que enviará uma mensagem para o e-mail informado com a senha do usuário mostrado na Figura 3.

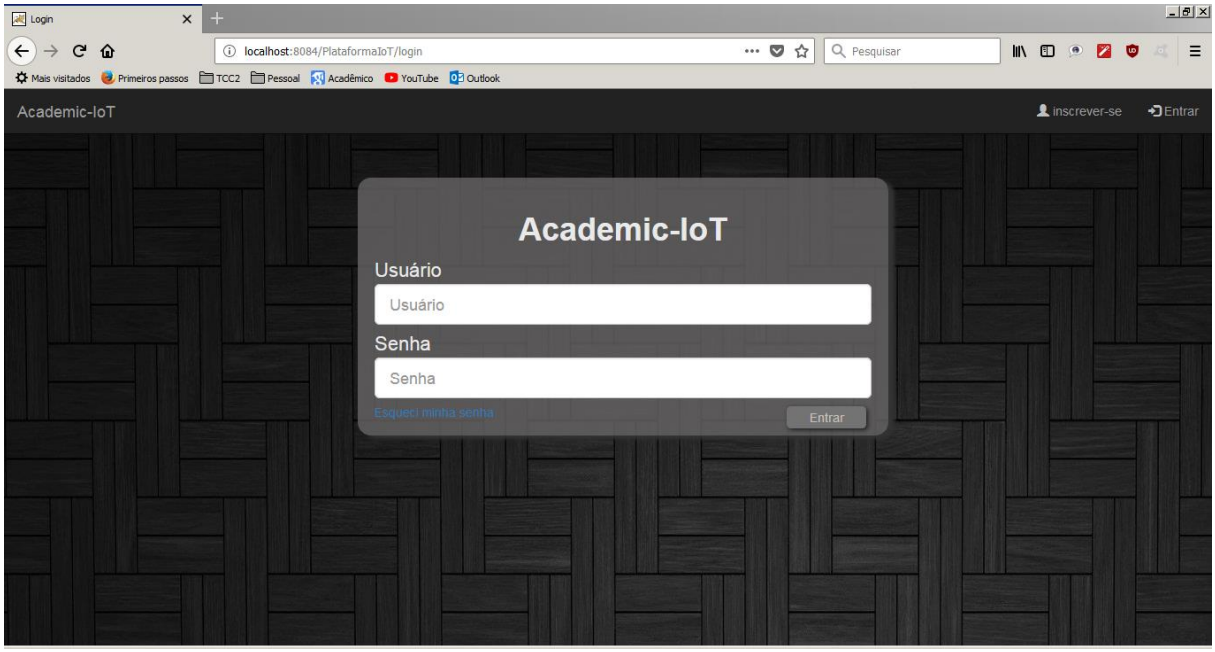

Figura 2 - Tela de login

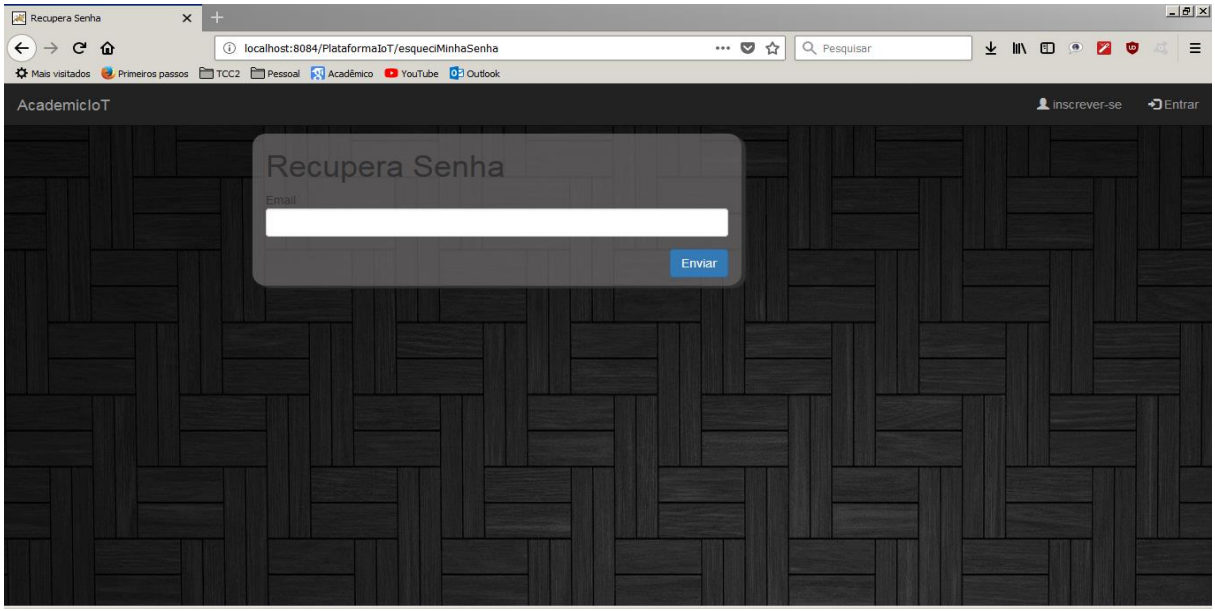

Figura 3 - Recupera senha

Caso o usuário não tenha cadastro, a plataforma oferece a opção de cadastro de um novo usuário, acessando o campo “inscrever-se”, que redireciona para pagina de cadastro de usuário é necessário informar o nome, o e-mail, um nome para *login*, uma senha para *login* e a repetição da mesma senha, para a fim de garantir que não haja erro de digitação mostrado na Figura 4.

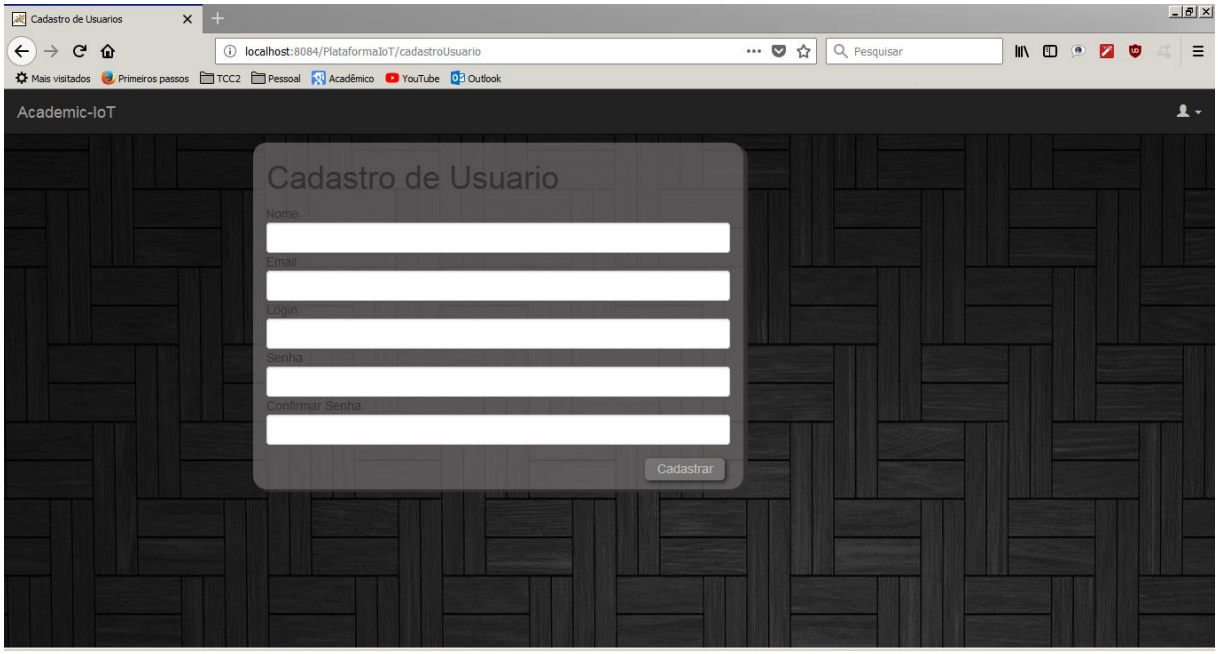

Figura 4 - Tela cadastro de usuário

4.4.3 Tela Principal

Após a validação dos dados do usuário, o usuário é direcionado tela principal da plataforma, apresentada na Figura 5. Nesta tela persiste uma barra superior que é fixa em todas as telas da plataforma, com um campo com o nome do usuário logado, onde é possível acessar a tela de edição de usuário acessando o campo “perfil”, onde é expresso na Figura 14, outra opção é sair da plataforma acessando o campo “sair”.

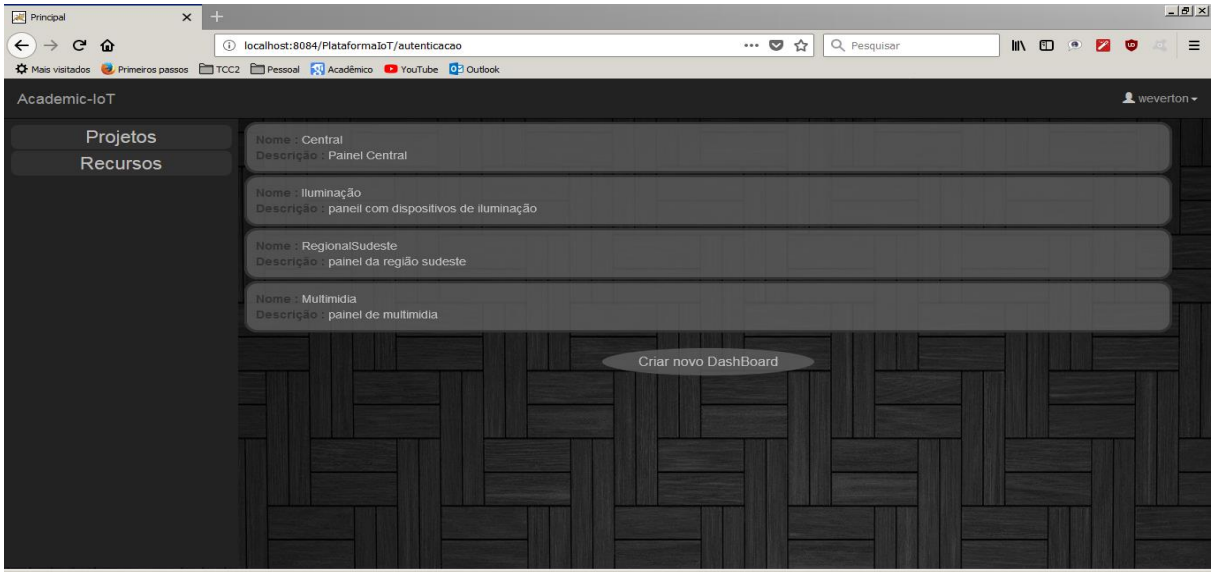

Figura 5 - Tela principal

Na tela principal é mostrada uma lista com os *dashboard's* do usuário, que quando clicado é exibido informações referentes aos recursos associados, como o nome do recurso e um gráfico com dados armazenados, coletado do recurso físico e/ou virtual, exposto na Figura 6.

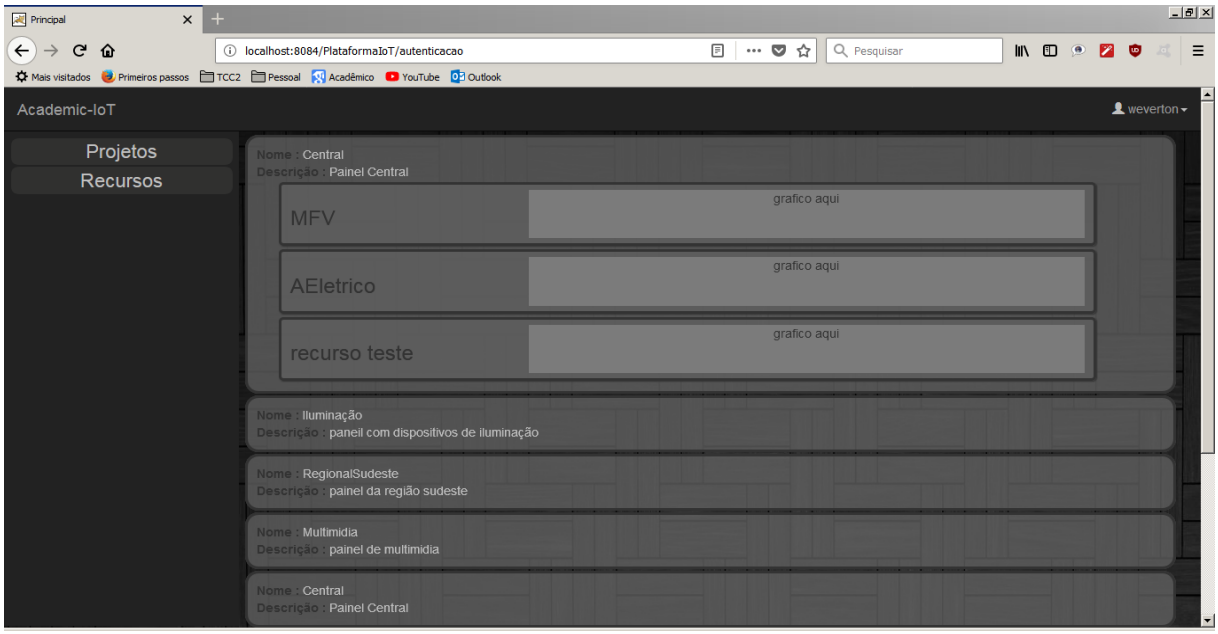

Figura 6 - Tela principal: Informação dasboard

Acessando o campo "Criar novo DashBoard" localizado no fim da listagem de *dashboard's*, direcionando a tela de cadastro de *dashboard*, onde será exibido um formulário expresso na Figura 7, para o preenchimento dos dados, possibilitando o cadastro de um novo *dashboard*, onde é necessário informar o nome e uma descrição do *dashboard*. Ao acessar o campo "Recurso", é apresentado uma listagem de recursos que estão vinculados a algum projeto do usuário logado, proporcionando a adicionar recurso(s) ao *dashboard*, bastando clicar em algum dos recursos listados, sendo mostrado em um painel ao lado direito do formulário de cadastro do *dashboard*, uma lista dos recursos selecionados, que serão vinculados ao *dashboard* à ser cadastrado, após a confirmação do cadastro.

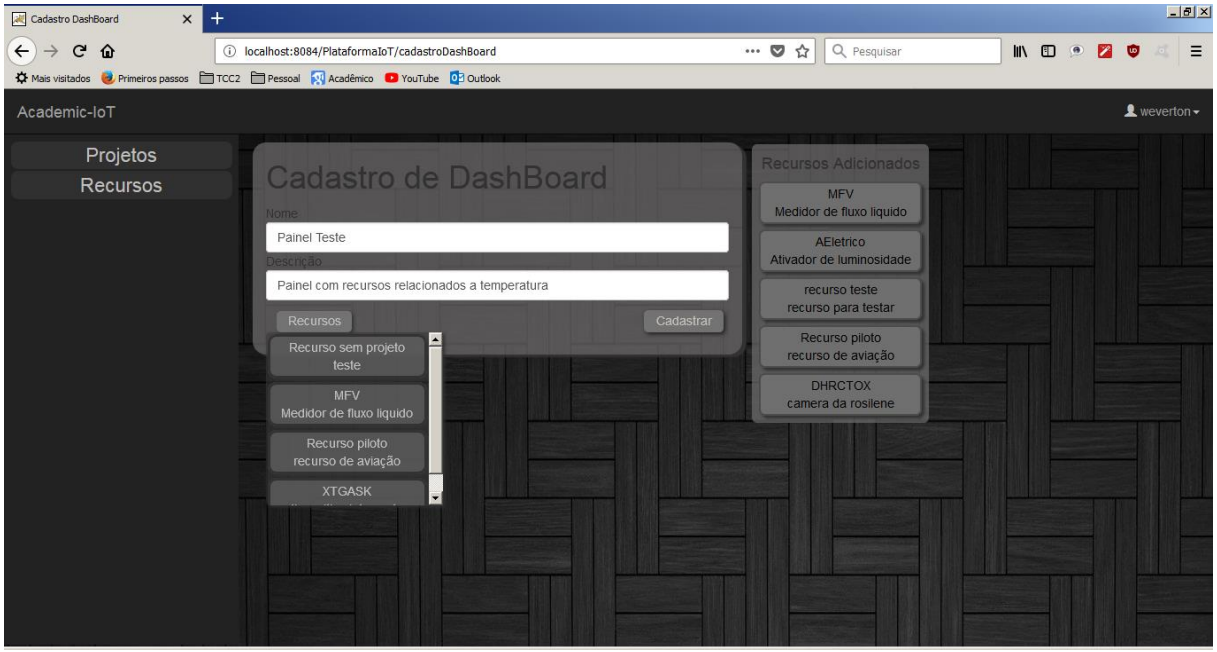

Figura 7 - Tela de cadastro de dashboard

Também é possível observar um painel, disposto verticalmente na lateral esquerda da tela principal exposto na Figura 5, onde são dispostos dois campos, o campo “Projetos” e o campo “Recursos”.

O campo “Projetos”, quando clicado, é listado todos os projetos do usuário, onde é possível exibir informações sobre o projeto. Para isso, basta clicar em um determinado projeto na listagem, onde serão exibidas informações do projeto selecionado, no qual é apresentado os campos: “Adicionar dispositivo” e “Recursos do Projeto” exibido na Figura 8.

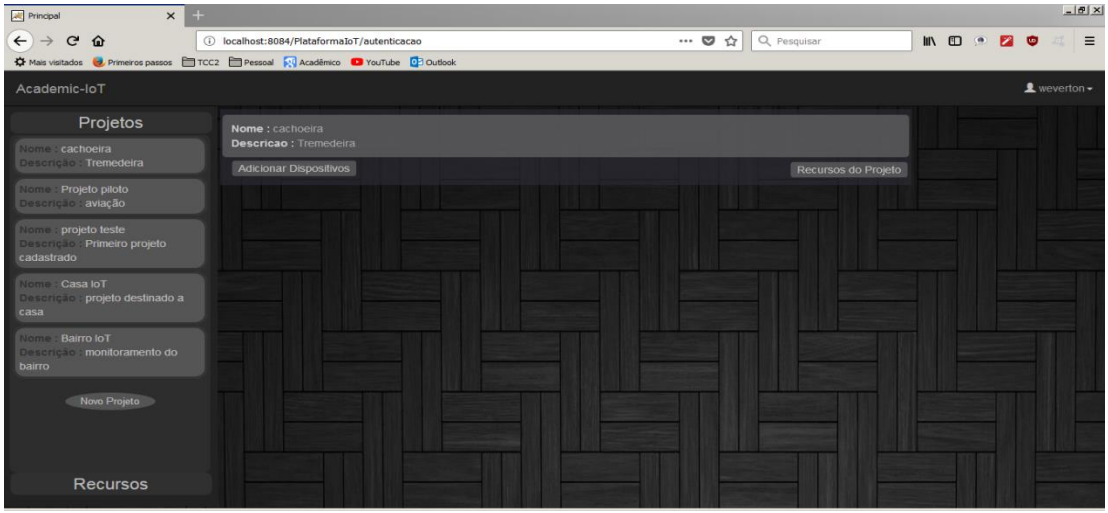

Figura 8 - Tela principal: Informações do projeto.

O campo “Adicionar Dispositivo”, quando clicado, é apresentado uma listagem de recursos que estão vinculados nenhum projeto, possibilitando adicionar recurso(s) ao projeto, bastando clicar em algum dos recursos listados como apresenta a Figura 9.

Figura 9 - Tela Principal: Informações do projeto2.

O campo “Recursos do Projeto”, quando clicado, exibido um painel com uma lista dos recursos associados ao projeto selecionado como mostra a Figura 10.

Figura 10 - Tela Principal: Informações do projeto3

Acessando o campo “Novo Projeto” localizado no fim da listagem de projeto, é direcionando a tela de cadastro de projeto mostrado na Figura 11, onde será exibido um formulário para a imputação dos dados, possibilitando o cadastro de um novo *projeto*, onde é necessário informar o nome e uma descrição do projeto. Ao acessar o campo ”Recursos”, é apresentada uma listagem de recursos que não estão vinculados a nenhum projeto, permitindo adicionar recurso(s) ao projeto, bastando clicar em algum dos recursos listados, sendo mostrado em um painel ao lado direito do formulário de cadastro do projeto, uma lista dos recursos selecionados, que serão vinculados ao projeto á ser cadastrado, após a confirmação do cadastro.

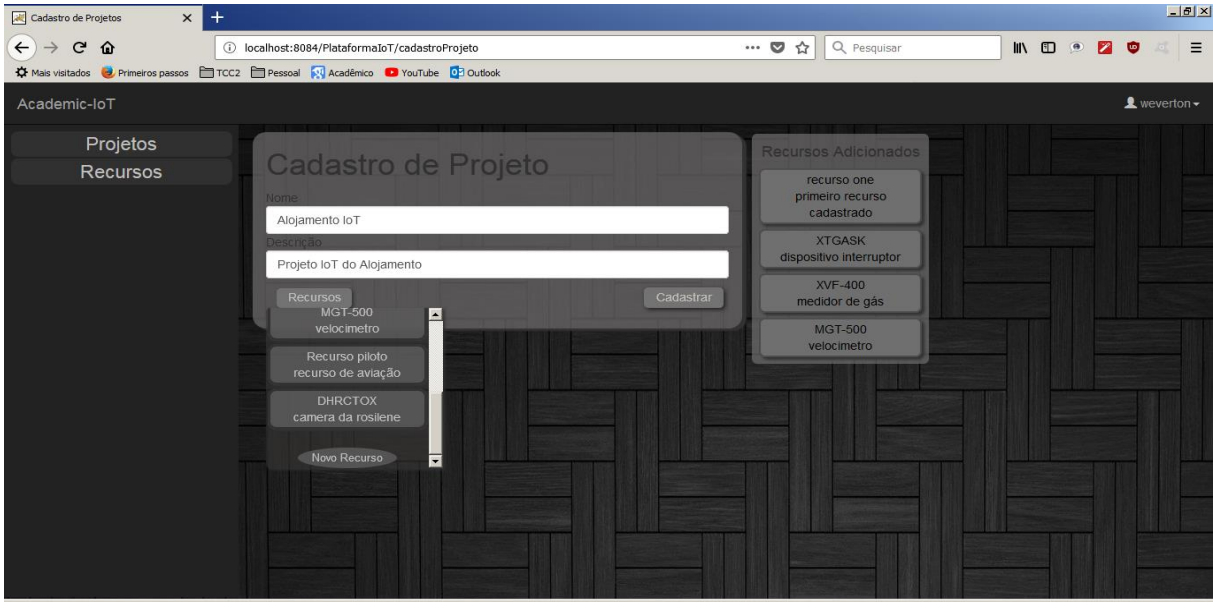

Figura 11 - Tela cadastro de projeto

O campo “Recursos” como mostra na Figura 5, localizado no painel, disposto verticalmente na lateral esquerda da tela principal, quando clicado, é listado todos os recursos do usuário, onde é possível exibir informações sobre o recurso assim como mostra a Figura 12. Para isso, bastando clicar em um determinado recurso da listagem, onde serão exibidas informações do recurso selecionado, como o nome, a descrição, o tipo de valor coletado e a categoria do dispositivo, também é exposto o campo “Dados do Recurso”, que quando clicado é apresentado um gráfico que contem informações coletadas do recurso físico/virtual.

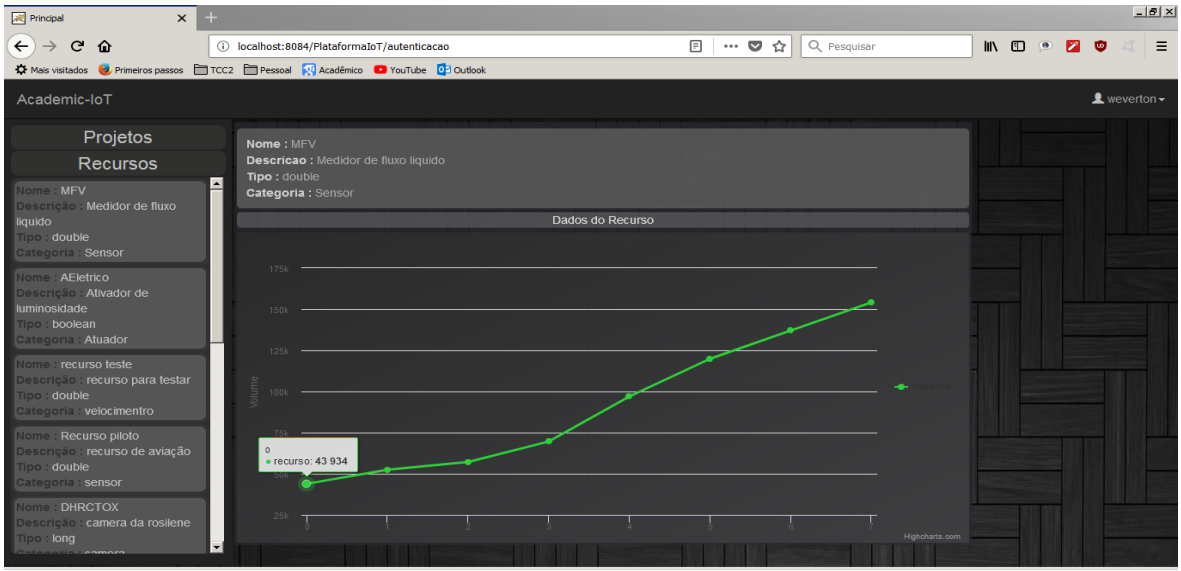

Figura 12 - Tela Principal: Informações do recurso.

Acessando o campo “Novo Recurso” localizado no fim da listagem de recursos, é direcionando a tela de cadastro de recurso, como mostra a Figura 13, onde será exibido um formulário para a imputação dos dados, possibilitando o cadastro de um novo recurso, onde é necessário informar o nome, uma descrição, o tipo de valor a ser coletado (*int*, *double*, *char* ou *boolean*) e a categoria do recurso, ao clicar no campo “Cadastrar”, um novo recurso é cadastrado e inicialmente não estará vinculado a nenhum projeto.

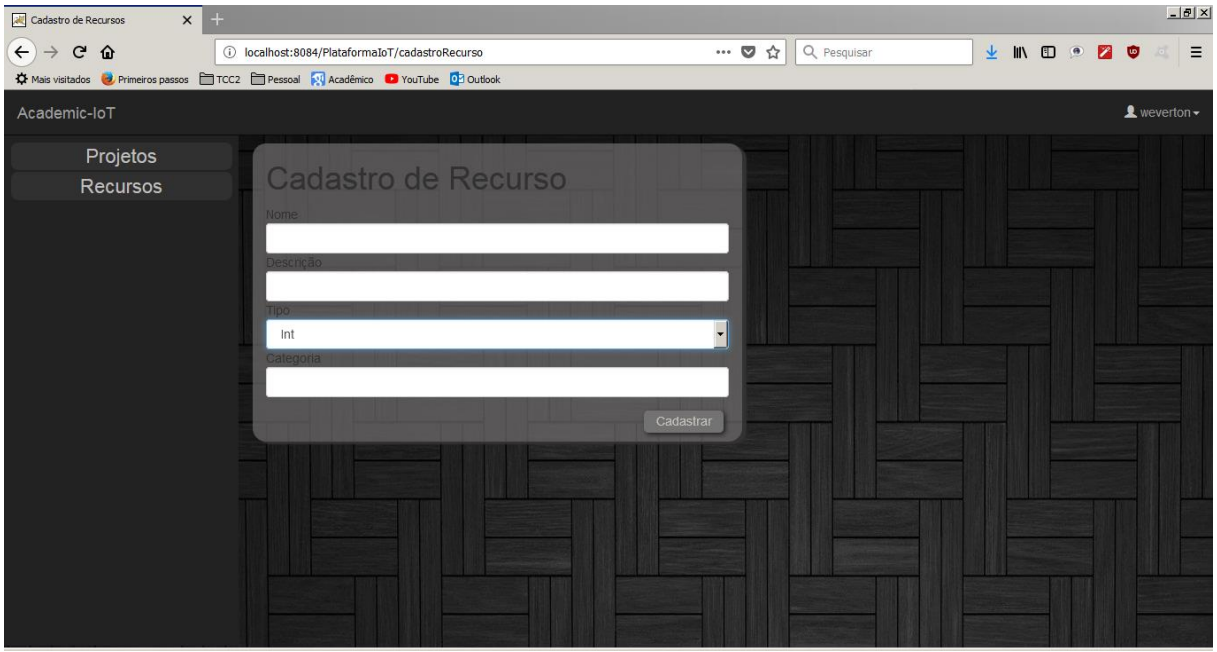

Figura 13 - Tela de cadastro de recurso

Na Tela principal canto direito superior mostrado na Figura 5, é apresentado um campo com o nome do usuário da sessão, ao clicar o usuário tem a opção de sair/*logout* da plataforma e a opção de acesso o perfil do usuário exposto na Figura 14, possibilitando a alteração das informações referentes a usuário.

Figura 14 - Tela de perfil do usuário

5. Conclusão

Esse trabalho apresentou o desenvolvimento de uma plataforma que possibilita a hospedagem de serviços relacionados à Internet das Coisas podendo receber e gerenciar as informações coletadas dos recursos.

O desenvolvimento da plataforma adotando ferramentas e *frameworks open-source*, embora sem investimento inicial, demandou um grande esforço para o aprendizado das novas tecnologias, devido à dificuldade em encontrar assuntos e materiais específicos.

Além disso, também foi possível o desenvolvimento de um protótipo funcional da plataforma, fato que proporcionou a aplicação prática dos conceitos assimilados. Entretanto, a disponibilidade de tempo limitada para a dedicação ao projeto não permitiu que fossem desenvolvidos todas as funcionalidades idealizadas, sendo concluído os essenciais e básicos para o uso da plataforma.

Contudo, melhorias são indispensáveis, como organização dos elementos nas telas, designer das telas e a adição de novos recursos, com a finalidade de aumentar a usabilidade da plataforma.

Como trabalhos futuros, podemos citar algumas melhorias como, por exemplo:

- Projeto compartilhado, permitindo que vários usuários tenham acesso a recursos cadastrados por outros usuários.
- Recursos públicos e privados, onde os recursos públicos podem ser visualizados e utilizados por outros usuários.
- Serviço de gatilho, a fim enviar mensagens de aviso sobre alteração em métricas pré-estabelecidas de recursos específicos após a leitura de seus dados.

6. Referências Bibliográfica

ANTUNES, J. B. et al. ManIoT: Uma Plataforma para Gerenciamento de Dispositivos da Internet das Coisas. 20 out. 2016.

BANAFA, A. *Internet of Things (IoT): The Third Wave*. 26 Jan. 2015. Disponível em: <https://www.bbvaopenmind.com/en/internet-of-things-iot-the-third-wave/>. Acesso em: 7 jul. 2017.

BALAGUER, A. L. A Internet das Coisas: das origens ao futuroStartupi, 27 out. 2014. Disponível em: <https://startupi.com.br/2014/10/internet-das-coisas-das-origens-ao-futuro/>. Acesso em: 5 jul. 2017

BANDYOPADHYAY, D.; SEN, J. *Internet of things: Applications and challenges in technology and standardization*. *Wireless Personal Communications*, v. 58, n. 1, p. 49–69, 2011.

DAVET, P. T. Plataformas para a IoT: Desafios e Projetos. 2015. Disponível em: <http://docplayer.com.br/19909439-Patricia-teixeira-davet-plataformas-para-a-iot-desafios-e-projetos.html>. Acesso em: 8 jul. 2017.

DE FRANÇA, T. C. et al. Web das coisas: conectando dispositivos físicos ao mundo digital. Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2015.

DELICATO, F. C., Pires, P. F., Batista, T., Cavalcante, E., Costa, B., Barros, T. “Towards an IoT ecosystem”. Proceedings of the First International Workshop on Software Engineering for Systems of Systems. USA, ACM, pp. 25-28. 2013.

GUINARD, D. et al. *Interacting with the SOA-Based Internet of Things: Discovery, Query, Selection, and On-Demand Provisioning of Web Services. IEEE Transactions on Services Computing*, v. 3, n. 3, p. 223–235, jul. 2010.

LI, T. et al. *A Storage Solution for Massive IoT Data Based on NoSQL. Proceedings of the 2012 IEEE International Conference on Green Computing and*

Communications. Anais...: GREENCOM ’12.Washington, DC, USA: IEEE *Computer Society*. 2012. Disponível em: <http://dx.doi.org/10.1109/GreenCom.2012.18>. Acesso em: 7 jul. 2017

Lang, J. P. (2014). Linksmart middleware platform portal. https://linksmart.eu/redmine/. Acessado em: 28 jul 2017.

LOUREIRO GARRIDO, R. Estudio plataformas IoT. B.S. thesis. Universitat Oberta de Catalunya, 2015.

MELO, S. IPv6: Portas abertas para a era da “Internet das Coisas” . 2012. Disponível em: <https://imasters.com.br/artigo/25326/tecnologia/ipv6-portas-abertas-para-a-era-da-internet-das-coisas?trace=1519021197&source=single>. Acesso em: 7 jul. 2017.

Moreiras, A. “IPv6, um desafio técnico para a Internet”, fev. 2014. Disponível em: <http://cio.com.br/tecnologia/2014/02/04/ipv6-um-desafio-tecnico-para-a-internet/>. Acesso em: 6 jul. 2017.

MOSQUITTO, MOSQUITTO, 12 abr. 2018. Disponível em: < https://mosquitto.org/>. Acesso em: 12 abr. 2018

MQTT, MQTT, 7 jul. 2017. Disponível em: <http://mqtt.org/>. Acesso em: 8 jul. 2017

PAHO, Eclipse Paho, 5 abr. 2018. Disponível em: <http://www.eclipse.org/paho/>. Acesso em: 5 abr. 2018

PEREIRA, V. R. F. Aplicação de *Internet of Things* em casas inteligentes: Serviços de Rede, 2014. Disponível em: <http://recipp.ipp.pt/handle/10400.22/7157>. Acesso em: 20 jun. 2017.

PIRES, P. F. et al. Plataformas para a Internet das Coisas. Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2015.

RAZZAQUE, M. A. et al. *Middleware for internet of things: a survey. IEEE Internet of Things Journal*, v. 3, n. 1, p. 70–95, 2016.

Santos, B. P. et al. Internet das Coisas: da Teoria à Prática p. 1-50, 2016. Disponível em: <http://homepages.dcc.ufmg.br/~bruno.ps/iot-tp-sbrc-2016/>. Acesso em: 6 jul. 2017.

SANTOS, V. A internet das coisas e a TAR: onde está o homem? Temática, v. 9, n. 10, 2014.

SCHEUNEMANN, D. A. Ciência de situação na IOT: uma arquitetura explorando processamento híbrido de contexto. 1 dez. 2016.

SILVA, T. B. DA et al. A INTERNET DAS COISAS: SERÁ A INTERNET DO FUTURO OU ESTÁ PRESTES A SE TORNAR A REALIDADE DO PRESENTE? Engenharias On-line, v. 1, n. 1, p. 41–50, 17 ago. 2015a.

SILVA, E. C. G. F. et al. Um Survey sobre Plataformas de Mediação de Dados para Internet das Coisas, 2015a. Disponível em: <https://search.datacite.org/works/10.13140/RG.2.1.3266.0325>. Acesso em: 6 jul. 2017.

VALENTE, B. A. L. Um *middleware* para a Internet das coisas. Universidade de Lisboa, 2011. Disponível em: <http://repositorio.ul.pt/handle/10451/9211>. Acesso em: 3 jul. 2017.

VEGA-BARBAS, M. et al. *Smart spaces and smart objects interoperability architecture (S3OiA). Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on. Anais...* [S.l.: s.n.], 2012. p.725–730.

VITERBO, J. Descoberta de Serviços em Ambientes de Computação Ubıqua. 2007.

ZYRIANOFF, I.; Borelli, F.; Kamienski, C.; “SenSE – *Sensor Simulation Environment*: Uma ferramenta para geração de tráfego IoT em larga escala”, 2017. SBRC 2017. Salão de Ferramentas.

7. ANEXOS

ANEXO A – Configuração do cliente MQTT

PlataformaIoT - NetBeans IDE 8.2

Arquivo Editar Exibir Navegar Código-Fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

```
28   * @author Weverton
29   */
30  public class ClienteMQTT implements MqttCallback {
31      static final String BROKER_URL = "127.0.0.1:1883";
32      static final String CLIENT = "125878";
    private ProcessadorTipo processador;
34      private MqttClient myClient;
35      private MqttConnectOptions connOpt;
36
37      public void conectar() {
38          connOpt = new MqttConnectOptions();
39          connOpt.setCleanSession(true);
40          connOpt.setKeepAliveInterval(30);
41          connOpt.setUserName("academicIoT");
42          connOpt.setPassword("sept1999Kevin".toCharArray());
43          try {// Connect to Broker
44              myClient = new MqttClient(BROKER_URL, CLIENT);
45              myClient.setCallback(this);
46              myClient.connect(connOpt);
47          } catch (MqttException e) {
                e.printStackTrace();
49          }
50      }
51
52      public void runClient() throws MqttException {
53          this.conectar();
54          try {
55              int subQoS = 0;
56              myClient.subscribe("/academicIoT/recursos/#", subQoS);
57              while(true) {
                    Thread.sleep(5000);
59              }
            } catch (Exception e) {
                e.printStackTrace();
62          } finally{
63              myClient.disconnect();
64          }
65      }
```