



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS QUIXADÁ**  
**BACHARELADO EM ENGENHARIA DE SOFTWARE**

**LUCAS SALES VIEIRA**

**REDUÇÃO DO CONSUMO DE ENERGIA DA UNIVERSIDADE FEDERAL DO  
CEARÁ - CAMPUS QUIXADÁ ATRAVÉS DE UM SERVIÇO SMART GREEN**

**QUIXADÁ – CEARÁ**

**2016**

LUCAS SALES VIEIRA

REDUÇÃO DO CONSUMO DE ENERGIA DA UNIVERSIDADE FEDERAL DO CEARÁ -  
CAMPUS QUIXADÁ ATRAVÉS DE UM SERVIÇO SMART GREEN

Monografia apresentada no curso de Engenharia de Software da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia de Software. Área de concentração: Computação.

Orientador: Prof. Dr. Marcio Espíndola Freire Maia

QUIXADÁ – CEARÁ

2016

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

V716r Vieira, Lucas Sales.

Redução do consumo de energia da Universidade Federal do Ceará - campus Quixadá através de um serviço Smart Green / Lucas Sales Vieira. – 2016.  
53 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Engenharia de Software, Quixadá, 2016.  
Orientação: Prof. Dr. Marcio Espíndola Freire Maia.

1. Computação em Nuvem. 2. Computação Ubíqua. 3. Sistemas embarcados (Computadores). I. Título.  
CDD 005.1

---

LUCAS SALES VIEIRA

REDUÇÃO DO CONSUMO DE ENERGIA DA UNIVERSIDADE FEDERAL DO CEARÁ -  
CAMPUS QUIXADÁ ATRAVÉS DE UM SERVIÇO SMART GREEN

Monografia apresentada no curso de Engenharia de Software da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Engenharia de Software. Área de concentração: Computação.

Aprovada em:

BANCA EXAMINADORA

---

Prof. Dr. Marcio Espíndola Freire Maia (Orientador)  
Campus Quixadá  
Universidade Federal do Ceará – UFC

---

Prof. Dr. Arthur de Castro Callado  
Campus Quixadá  
Universidade Federal do Ceará - UFC

---

Prof. Msc. Bruno Góis Mateus  
Campus Quixadá  
Universidade Federal do Ceará - UFC

A Deus,

A minha família, amigos e a todos que não vão  
com a minha cara.

## **AGRADECIMENTOS**

Primeiramente a Deus depois à minha mãe Nubian Sales Vieira e meu pai José Amaurilio Carvalho Vieira, aos meus tios Maria do Socorro Carvalho Vieira e Cristiano Carvalho Vieira.

Ao Prof. Dr. Marcio Espíndola Freire Maia, por acreditar em mim e me dar todo o apoio para a conclusão deste trabalho de curso.

Aos professores participantes da banca examinadora Prof. Dr. Arthur de Castro Callado e Prof. Me. Bruno Góis Mateus pelo tempo, pelas valiosas colaborações e sugestões.

Agradeço a todas as amigas verdadeiras que construí durante esses anos, mas gostaria de agradecer de forma especial ao Carlos Matheus, Robson Cavalcante, Carlos Eduardo, Luan Lima, Bruno Barreto, Jacques Nier(Apodi), Randerson Lessa, Rubéns Lima, Pedro Sávio, Matheus Souza, Isabelle Oliveira e Tamiris Costa por ter marcado minha vida de uma forma positiva ou contribuído para este trabalho.

Agradeço a família do seu Gomes por ter me recebido, ajudado e apoiado quando cheguei em Quixadá. Agradeço a Helena (Pois sei que ela não gosta de ser chamada de dona ou senhora) por ter me ajudado em alguns momentos delicados durante meu curso e ter me motivado.

Gostaria de agradecer aos servidores do campus Quixadá em especial ao Pipah e o Kaká por terem me ajudado durante este trabalho.

“As coisas mudam. Não significa que melhoram.  
Você precisa tornar as coisas melhores. Não  
pode ficar falando e esperar o melhor.”

(Dr. House)

## RESUMO

Atualmente, o número de dispositivos conectados à internet já ultrapassa a população humana, e com isso áreas como a IOT (*Internet of Things*) vem crescendo cada vez mais. Nesse contexto, com o aumento da população e dos dispositivos, o aumento da poluição e a diminuição dos recursos naturais tomando em grandes proporções. Com a intenção de tentar diminuir o consumo de energia e ajudar a facilitar o trabalho dos servidores da Universidade Federal do Ceará - Campus Quixadá através da *Internet of Things* (IOT), este trabalho mostra o desenvolvimento de um serviço para controlar as lâmpadas, condicionadores de ar e projetores de uma sala de aula, fornecendo as funcionalidades de ligar e desligar esses objetos e permitindo controlar a temperatura dos condicionadores de ar, diminuindo o consumo de energia da instituição.

**Palavras-chave:** Computação em Nuvem. Computação Ubíqua. Sistemas Embarcados

## **ABSTRACT**

Currently, the number of devices connected to the Internet exceeds the human population, increasing the importance of research areas like IOT (Internet of Things). In this context, with the growth in the population has led to expansion in the pollution and to a contraction in the available natural resources. Trying to reduce the energy consumption and to help the work of users of the Federal University of Ceará - Quixadá Campus, this work shows the development of a service to control lamps, air conditioners and projectors of a classroom, providing functionalities to turn on and off these objects and to permit to control the temperature of the air conditioners.

**Keywords:** Cloud Computing. Ubiquitous Computing. Embedded Systems

## LISTA DE FIGURAS

Figura 1 – Arquitetura Referencial . . . . .	17
Figura 2 – Publish/Subscribe . . . . .	19
Figura 3 – Arduino Uno . . . . .	24
Figura 4 – Representação Arquitetura do Sistema . . . . .	30
Figura 5 – Ligar/Desligar um objeto . . . . .	34
Figura 6 – Diagrama de classe do sistema . . . . .	35
Figura 7 – Diagrama de sequência exemplificando a persistência dos dados . . . . .	36
Figura 8 – Diagrama de Atividade . . . . .	39
Figura 9 – Ligar/Desligar um objeto . . . . .	39
Figura 10 – Tela login . . . . .	48
Figura 11 – Cadastro de Bloco . . . . .	48
Figura 12 – Cadastro de Sala . . . . .	49
Figura 13 – Listagem de Blocos . . . . .	49
Figura 14 – Listagem de Salas . . . . .	50
Figura 15 – Listagem de Salas . . . . .	50
Figura 16 – Funcionalidade de Programar Objetos . . . . .	51

## LISTA DE QUADROS

Quadro 1 – Conceitos . . . . .	21
Quadro 2 – Vantagens Arduino . . . . .	24
Quadro 3 – Características da Computação Nuvem . . . . .	25
Quadro 4 – Resultados da Análise . . . . .	41

## LISTA DE ABREVIATURAS E SIGLAS

CoAP	Constrained Application Protocol
CSS	Cascading Style Sheets
DaaS	Database as a Service
EC2	Elastic Compute Cloud
EEPROM	Electrically Erasable Programmable Read Only Memory
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
IOT	Internet of Things
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
KWh	Quilowatt hora
LDAP	Lightweight Directory Access Protocol
MQTT	Message Queue Telemetry Transport
PaaS	Plataform as a Service
PC	Computador Pessoal
RAM	Random Access Memory
RFID	Radio Frequency IDentification
SaaS	Software as a Service
SGBD	Sistema de Gerenciamento de Banco de Dados
UDP	User Datagram Protoco

VPN      Virtual Private Network

W      Watt

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>16</b>
<b>2.1</b>	<b>Internet das Coisas (IOT)</b>	<b>16</b>
<b>2.1.1</b>	<i>Arquitetura de Referência</i>	<b>17</b>
<b>2.1.2</b>	<i>Protocolos para IOT</i>	<b>19</b>
<b>2.1.2.1</b>	<i>Message Queue Telemetry Transport (MQTT)</i>	<b>19</b>
<b>2.1.2.2</b>	<i>Constrained Application Protocol (CoAP)</i>	<b>20</b>
<b>2.2</b>	<b>Smart Campus e Green Campus</b>	<b>21</b>
<b>2.3</b>	<b>Sistemas Embarcados</b>	<b>22</b>
<b>2.3.0.1</b>	<i>Arduino</i>	<b>23</b>
<b>2.4</b>	<b>Computação em Nuvem</b>	<b>24</b>
<b>2.4.1</b>	<i>Serviços oferecidos pela Computação em Nuvem</i>	<b>25</b>
<b>2.4.2</b>	<i>Tipos de Nuvem</i>	<b>26</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>27</b>
<b>3.1</b>	<b>Towards the Implementation of IoT for Environmental Condition Monitoring in Homes</b>	<b>27</b>
<b>3.2</b>	<b>Constructing the Green Campus within the Internet of Things Architecture</b>	<b>28</b>
<b>3.3</b>	<b>Ubiquitous Smart Home System Using Android Application</b>	<b>28</b>
<b>4</b>	<b>PROPOSTA</b>	<b>30</b>
<b>4.1</b>	<b>Arquitetura do Sistema</b>	<b>30</b>
<b>4.1.1</b>	<i>Cliente</i>	<b>31</b>
<b>4.1.2</b>	<i>Serviço Web</i>	<b>31</b>
<b>4.1.3</b>	<i>Intermediário</i>	<b>31</b>
<b>4.1.4</b>	<i>Persistência</i>	<b>31</b>
<b>4.1.5</b>	<i>Placa</i>	<b>32</b>
<b>4.1.6</b>	<i>Visão geral das Funcionalidades</i>	<b>33</b>
<b>4.2</b>	<b>Funcionalidades do Sistema</b>	<b>34</b>
<b>4.2.1</b>	<i>CRUD</i>	<b>35</b>
<b>4.2.2</b>	<i>Controle dos Objetos</i>	<b>37</b>
<b>5</b>	<b>RESULTADOS E DISCUSSÕES</b>	<b>40</b>

<b>5.1</b>	<b>Avaliação e Resultados . . . . .</b>	<b>40</b>
<b>5.2</b>	<b>Problemas . . . . .</b>	<b>41</b>
<b>5.3</b>	<b>Considerações sobre os resultados . . . . .</b>	<b>42</b>
<b>6</b>	<b>TRABALHOS FUTUROS . . . . .</b>	<b>43</b>
<b>7</b>	<b>CONSIDERAÇÕES FINAIS . . . . .</b>	<b>44</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>45</b>
	<b>APÊNDICE A – Imagens do Sistema . . . . .</b>	<b>48</b>

## 1 INTRODUÇÃO

A Internet das Coisas (*Internet of Things*- IOT) vem ganhando espaço recentemente, tanto no cenário industrial como no cenário acadêmico, por causa da sua ideia central de que todos os objetos ou *things* (coisas) estejam conectados à Internet e que seja necessário o mínimo de ação das pessoas sobre os objetos. A IOT propõe criar um mundo melhor para o ser humano onde os objetos saibam o que queremos e que eles sejam capazes de aprender sem a necessidade de instruções (PERERA et al., 2014). Segundo Lima (2014), em 2003, o número de dispositivos conectados à internet era de cerca de 500 milhões para cerca de 6.3 bilhões de pessoas. Em 2008, esse número aumentou de uma forma que chegou a ultrapassar a população humana e, em 2012, eram mais de 8.7 bilhões de dispositivos conectados à internet para cerca de 7 bilhões de pessoas no mundo. Segundo Zafari, Papapanagiotou e Christidis (2016), estima-se que em 2020 esse número chegue a 24 bilhões.

A IOT fornece várias possibilidades de atuação em cenários como casas, prédios, campi, cidades, o que abre uma grande possibilidade de ideias de aplicações para serem desenvolvidas (ATZORI; IERA; MORABITO, 2010). De acordo com estes autores, é possível agrupar esses domínios em, Transporte e logística de domínio, domínio de Saúde, ambiente inteligente, pessoal e área social. Este trabalho irá se aprofundar no domínio de *Smart Environment* (Ambientes Inteligentes). Neles, existe uma troca de dados entre dispositivos, possivelmente utilizando a internet, permitindo que recursos como luzes, portas ou projetores possam ser acessados e controlados (NIE, 2013).

Neste contexto, com o aumento da população e os recursos naturais diminuindo, os governos implantam medidas para reduzir a liberação de carbono e a diminuição dos gastos de energia. Como exemplo concentração de CO<sub>2</sub> aumentou 25% entre 1955 e 2013, isso trouxe várias consequências ao nosso planeta como chuva ácida e erosão dos solos (WANG, 2014).

Este trabalho propõe a modelagem de um serviço que utilize dispositivos embarcados acessíveis através de web service para controlar luzes, condicionadores de ar e projetor das salas de aulas da Universidade Federal do Ceará (UFC) - Campus Quixadá. Esses dispositivos embarcados receberão comandos de um sistema web que enviará comandos para as luzes, condicionadores de ar e projetores das salas. A criação desse serviço tem como objetivo ajudar a diminuir o consumo de energia do Campus, além de ajudar nas aulas e ajudar a facilitar o trabalho dos servidores da instituição. Esse trabalho irá beneficiar professores, alunos, funcionários e o Campus através do controle de energia.

O trabalho está dividido da seguinte forma: 2-Fundamentação Teórica, com uma discussão sobre os temas mais relevantes do trabalho e como eles são aplicados no mesmo; 3-Trabalhos Relacionados, com trabalhos que inspiram a realização deste; 4-Proposta, explica como o serviço foi desenvolvido e sua arquitetura; 5-Resultados e Discussão, com os resultados da aplicação do serviço e uma discussão sobre os resultados; 6-Trabalhos Futuros, com uma discussão sobre possíveis evoluções deste trabalho e por ultimo 7-Considerações Finais com a conclusão deste trabalho.

## 2 FUNDAMENTAÇÃO TEÓRICA

Na fundamentação teórica serão descritos os conceitos necessários para o entendimento deste trabalho. A primeira seção aborda Internet das Coisas e a sua importância para o trabalho. A segunda seção trata de *Smart Campus* (Campus Inteligente) e *Green Campus* (Campus Verde) e a terceira seção aborda sistemas embarcados.

### 2.1 Internet das Coisas (IOT)

O termo Internet das Coisas (IOT) foi utilizado pela primeira vez por Kevin Ashton em 1999 e com o passar dos anos, ganhou força e espaço em diferentes áreas como e-health, casas e prédios inteligentes, ou sistemas de monitoramento de tráfego (GUBBI et al., 2013). Internet das Coisas é utilizado para representar uma extensão da Internet para objetos físicos (Coisas). Assim, ela é a comunicação de objetos, dispositivos pessoais e computadores através da Internet (MIORANDI et al., 2012).

A Internet das Coisas tem o objetivo de permitir a criação de novos serviços e novas abordagens para os serviços existentes (MIORANDI et al., 2012). Nela, os objetos aprendem com as atividades do usuário, criando assim objetos inteligentes. Um exemplo de objeto inteligente é a *Mother Sense*, que é um sistema para monitoramento de objetos físicos, sendo capaz de oferecer detalhes sobre as atividades dos objetos monitorados, além de enviar notificações, lembretes e alertas (SEN.SE, 2016).

De acordo com Gubbi et al. (2013), existem quatro domínios de aplicações da IOT: pessoal/doméstico, empresarial, utilidade e mobilidade. O domínio pessoal/doméstico é utilizado em sua maioria por pessoas que estão na mesma rede e utilizam a rede como principal fonte de comunicação. Um exemplo de uso desse domínio que está sendo estudado é o sistema de monitoramento de pacientes e idosos em suas casas (GUBBI et al., 2013). O domínio empresarial monitora e controla dispositivos em uma empresa, com restrições importantes quanto a privacidade dos dados coletados. O domínio de utilidade é usado normalmente para monitorar e melhorar serviços. Um exemplo de uso desse domínio é o monitoramento de pontos de eletricidade de uma casa utilizando o estudo desses pontos para melhorar o consumo de energia. Por fim, o domínio de mobilidade é focado no tráfego urbano e este domínio é um dos principais causadores de poluições sonoras e do ar. Um exemplo onde a IOT atua neste domínio na coleta de informações do tráfego a serem mostradas a seus usuário.

Este trabalho está inserido no domínio de IOT pessoal/doméstico, onde luzes, projetores e condicionadores de ar se comunicam com um dispositivo embarcado, que por sua vez recebe comandos de smartphones e do sistema web para realizar as ações como ligar e desligar as luzes.

### 2.1.1 Arquitetura de Referência

A arquitetura de referência ilustrada na **Figura 1** fornece uma descrição dos serviços presentes em um sistema, assim como as relações entre esses elementos e os princípios que definem essas relações.

Figura 1 – Arquitetura Referencial



Fonte: Adaptado de Holler et al. (2014)

As camadas dessa arquitetura serão discutidas a seguir. É importante ressaltar que as camadas não necessariamente se comunicam apenas com a camada que está abaixo dela de acordo com a **Figura 1**, podendo se comunicar com qualquer outra.

**Camada de Ativos:** Nessa camada estão os objetos que serão monitorados ou controlados através do serviço (HOLLER et al., 2014). Os objetos que serão controlados neste trabalho que são: as lâmpadas, condicionadores de ar e projetores se encontram nesta camada.

**Camadas de Recurso:** Nessa camada ficam os sensores, atuadores e *gateways* de vários tipos que fornecem agregação ou outros recursos que estão intimamente ligados com os recursos básicos. Essa camada permite a comunicação com a Camada de Ativos através de diversos modos como RFID e QR (HOLLER et al., 2014). É nessa camada que se encontra o Arduino que é o responsável por receber mensagens da Camada de Comunicação e controlar os

objetos na Camada de Ativos.

**Camada de Comunicação:** Essa camada é responsável por realizar a comunicação de uma extremidade que hospeda e executa os serviços com a outra extremidade que tem a lógica da aplicação. A comunicação é feita de acordo com a forma que cada camada aceita receber a comunicação. Normalmente em indústrias de redes móveis a comunicação é feita utilizando M2M (HOLLER et al., 2014). Neste trabalho a Camada de Comunicação é responsável por receber as informações da Camada de Aplicação e enviar essas informações para a camada de Recurso.

**Camada de Serviço:** Nessa camada ficam os servidores, *data center* e ambientes em nuvem (HOLLER et al., 2014). Ela é responsável por hospedar os serviços do sistema.

**Camada de Dados e Informação:** O principal foco dessa camada é organizar as informações, ou seja, dar suporte à lógica de controle (HOLLER et al., 2014). É nessa camada que será especificado como é a troca de mensagens do sistema, como o cliente se comunica com o servidor e como o servidor se comunica com a parte do sistema embarcada. É nessa camada que os protocolos de comunicação que são utilizados ficam.

**Camada de Aplicação:** Essa camada fornece aplicações para a IOT (HOLLER et al., 2014). É nesta camada que o sistema web se encontra. É através dessa camada que o usuário consegue se comunicar com os objetos das salas.

**Camada de Negócio:** Essa camada é focada em apoiar os valores de negócio dos *stakeholders* (HOLLER et al., 2014). Nessa camada ocorre a junção dos valores de negócio do *stakeholders* com a IOT.

**Camada de Segurança** É a camada responsável por garantir a segurança das informações. Essa camada é implementada em outras camadas que tem o seu nível de segurança (HOLLER et al., 2014).

**Camada de Configuração:** É nessa camada que é feita a gestão da solução do sistema. Ela é responsável pela operação, manutenção e administração do sistema (HOLLER et al., 2014). É nessa camada que se encontra o gerenciamento do serviço proposto neste trabalho.

Neste trabalho, utilizamos essa arquitetura de referência como base para desenvolver o serviço proposto. É importante ressaltar que ela prevê que pode não ser necessário utilizar todas as suas camadas e que parte de algumas dessas camadas podem ser implementadas dentro de outras, como é o caso da Camada de Segurança. A arquitetura deste trabalho, que não utilizou todas as camadas, será discutida na seção 4.1.

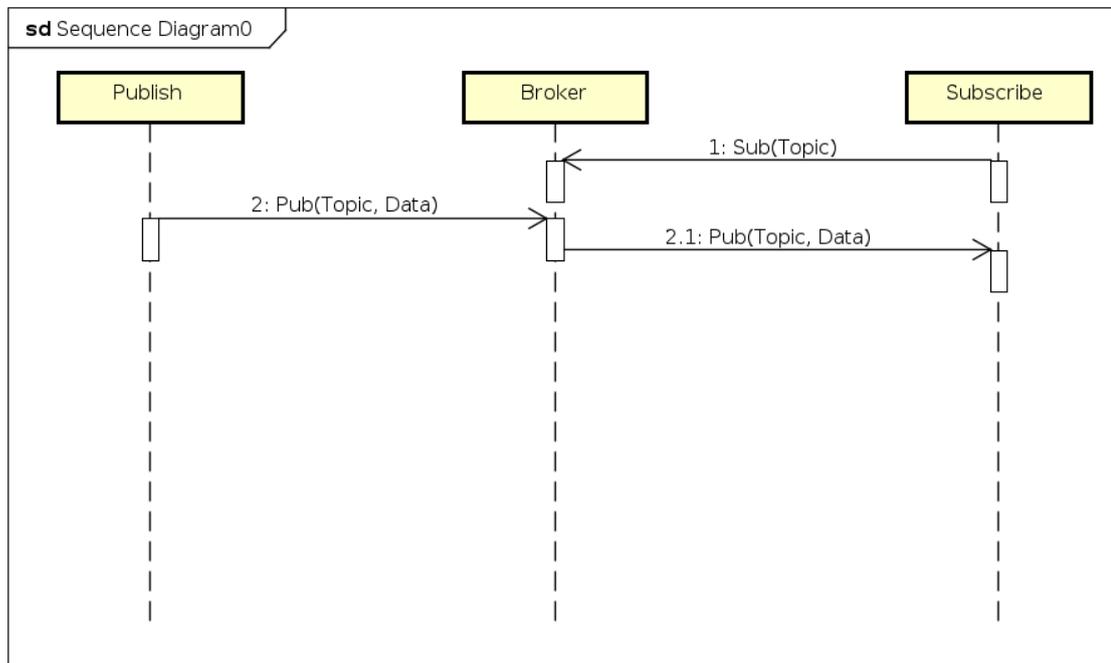
## 2.1.2 Protocolos para IOT

Nessa subseção iremos discutir sobre os dois protocolos mais utilizados atualmente em IOT e qual deles usaremos neste trabalho.

### 2.1.2.1 Message Queue Telemetry Transport (MQTT)

O MQTT é um protocolo desenvolvido pela IBM e é utilizado para realizar a comunicação entre máquinas na IOT. Ele foi projetado para ser um protocolo leve de trocas de mensagens baseada no padrão arquitetural *publish/subscriber* (MQTT, 2016a). Neste padrão, um elemento que deseja receber mensagens de um determinado tipo precisa se inscrever em um *broker*. Um segundo elemento publica mensagens no *broker* descrevendo o tipo da mensagem. O *broker* é responsável por encaminhar mensagens para os elementos interessados em um determinado tipo (EMBARCADOS, 2016). A **Figura 2** ilustra como funciona a troca de mensagens.

Figura 2 – Publish/Subscribe



powered by Astah

Fonte: Elaborada pelo Autor

O MQTT oferece três tipos de qualidade de serviço (*Quality of Service, QoS*) (MQTT, 2016b):

QoS 0: Neste serviço não há garantia de entrega da mensagem e nem mensagem de retorno se caso a mensagem chegue ou não .

QoS 1: Este serviço ele garante que a mensagem chegue pelo menos uma vez mas pode haver duplicação.

QoS 2: Este é o nível mais alto da qualidade de serviço do MQTT. Nele não há duplicações de mensagem e há uma garantia de entrega da mensagem.

#### 2.1.2.2 *Constrained Application Protocol (CoAP)*

O CoAP é um protocolo baseado no *Hyper Text Transfer Protocol Secure* (HTTP) que segue o modelo REST (*Representational State Transfer*). Os clientes conseguem acessar os recursos através dos métodos *GET*, *PUT*, *POST* e *DELETE*. Ele suporta a troca dados JSON, XML ou qualquer outro formato (COAP, 2016). Embora o CoAP seja parecido com o HTTP ele utiliza menos largura de banda e sua implementação é mais simples do que a do HTTP (KOVATSCH; DUQUENNOY; DUNKELS, 2011).

O CoAP, assim como o protocolo UDP (*User Datagram Protocol*), utiliza o transporte orientado a *datagramas* para realizar a troca de mensagens (SHELBY; HARTKE; BORMANN, 2014). Ele tem um mecanismo que permite que tipos diferentes de mensagens sejam utilizados para criar uma troca de mensagem entre o cliente e o servidor. elas podem ser:

**Confirmável:** As mensagens sempre enviam uma requisição ou uma resposta e exigem um reconhecimento (KULADINITHI et al., 2011)

**Não Confirmável:** É utilizado para mensagens que são utilizadas com frequência e não requerem um reconhecimento (KULADINITHI et al., 2011).

**Reconhecimento:** São mensagens de reconhecimento das mensagens do tipo "*Confirmável*", essas mensagens podem ser vazias ou conter uma resposta (KULADINITHI et al., 2011).

**Reset:** São mensagens que são enviadas caso as mensagens do tipo "*Confirmável*" não tenha sido entregues (KULADINITHI et al., 2011).

Neste trabalho será utilizado o protocolo MQTT para realizar a comunicação do sistema embarcado com o sistema web.

## 2.2 Smart Campus e Green Campus

Nie (2013) afirma que os *Smart Campus* (Campi inteligentes) surgiram graças ao avanço de IOT e Computação em Nuvem. Os Campi Inteligentes têm características próprias que, de acordo com Hirsch e Ng (2011), podem ser chamados de iLearning, iSocial, iGreen, iHealth, iManagement e iGovernance. Cada uma dessas características serve para garantir o bom funcionamento do Campus Inteligente.

Quadro 1 – Conceitos

Características	Conceitos
iLearning	Significa fornecer meios para que seja possível aprender sozinho ou em conjunto a qualquer hora e em qualquer lugar. A computação em Nuvem é uma das possíveis soluções para implementar essa característica.
iSocial	É voltada para a comunicação informal dentro do campus através do uso da internet e essas características se concentram em três áreas: no currículo do campus, extracurriculares e atividades sociais no geral.
iGreen	Tem a ver com a sustentabilidade, controle de gastos, gestão de recursos ociosos.
iHealth	Provê o controle de sistemas de saúde, alertas de epidemias e mostrar como se prevenir de alguma doença.
iManagement	É voltada mais para aspectos físicos como controle da segurança e monitoramento de prédios.
iGovernance	É a parte de fornecer o gerenciamento da parte organizacional do campus incluindo, gestão de processos e gestão da administração.

Fonte: Hirsch e Ng (2011).(com adaptações)

A *Green Campus* veio com o intuito de ajudar a computação a não desperdiçar recursos ociosos, a otimizar o uso de recursos para que não haja recursos ociosos ou, se houver, que seja o mínimo possível. Os *Smart Campus*, ou Campus Inteligente, usam a IOT para controlar objetos, fazendo com que os objetos sejam “inteligentes” aprendam e possam saber se estão ou não com recursos ociosos.

Com o crescente aumento da poluição no globo terrestre e os gastos de recursos naturais do nosso planeta sendo utilizados de forma desenfreada e não sustentável, os governos de todo o mundo criam formas de conter essas atividades e propor soluções para elas como o desenvolvimento sustentável (WANG, 2014). Com isso, o *Green Campus* tem como objetivo diminuir os gastos desnecessários ajudando assim a melhorar o meio ambiente.

Como as instalações de um Campus Inteligente são monitoradas e controladas para evitar o consumo desnecessário de energia e recursos ociosos, logo isso é visto também como a

construção de de um *Green Campus*.

Neste trabalho os conceitos que serão priorizados no desenvolvimento da arquitetura serão *iGreen* e *iManagement*, já que o objetivo desse trabalho é criar uma arquitetura que permita realizar o controle de luzes, projetores e condicionadores de ar.

### 2.3 Sistemas Embarcados

Atualmente os sistemas embarcados estão presentes em algumas atividades que realizamos e a sua presença vem crescendo a cada dia. Alguns exemplos que podemos ver onde eles estão presentes são os sistemas de carros e ônibus, fornos de micro-ondas com controle de temperatura inteligente e máquinas de lavar (CARRO; WAGNER, 2003). Eles são sistemas independentes que realizam tarefas que foram pré-determinadas (CUNHA, 2007).

Criar essa tecnologia não é fácil, pois envolve vários conceitos de computação que são complexos como o limite de consumo de potência sem perda de desempenho, baixa disponibilidade de memória, portabilidade e confiabilidade. Existem sistemas embarcados que tem uma estrutura que permite a ligação de componentes elas podem ser chamadas de *shields* . Alguns sistemas embarcados permitem que os usuários finais interajam com eles através de interfaces como teclados e *displays* (CARRO; WAGNER, 2003).

Segundo Cunha (2007), existem quatro tipos de aplicações dos sistemas embarcados: (1) propósito geral que nada mais são do que aplicações que se parecem com computadores de mesa e geralmente a interação do usuário com ela é feita através de terminal de vídeo ou monitores; (2) sistemas de controle de malha fechada em que a realimentação é realizada em tempo de execução normalmente os usuários não realizam interação com ele; (3) processamento de sinais: eles são responsáveis por realizar um grande processamento de informações em um intervalo curto de tempo e (4) comunicações de rede são os responsáveis por realizar a distribuição das informações e chaveamento, um exemplo são sistemas de telefonia.

Os sistemas embarcados funcionam de dois modos: reativo e controle em tempo real. O modo reativo funciona de acordo com eventos externos, como ao apertar um botão ou sistemas de controle de *loop*. Ele normalmente não tem um tempo específico para que os dados de entrada sejam enviados, mas logo após o envio dos dados ele deve executar a ação que foi pré-determinada na sua codificação. Por outro lado, no modo de controle em tempo real existe um tempo limite para que a ação seja executada, eles não dependem da entrada de dados para realizar suas ações e são capazes de realizar ações mesmo que não recebam dados de entradas

(CUNHA, 2007).

Os sistemas embarcados que realizam o controle de tempo real podem ser classificados em dois tipos *Soft Real Time* e *Hard Real Time*. No *Soft Real Time* as tarefas são executadas em um determinado tempo e caso a tarefa não seja cumprida não haverá graves consequências. Já os *Hard Real Time* as tarefas tem a obrigação de serem executadas no tempo determinado e caso ocorra algum erro, há graves consequências (CUNHA, 2007).

O modo de funcionamento do sistema embarcado que será utilizado nesse trabalho será do tipo reativo, pois ele tem que responder a eventos (ligar e desligar luzes, por exemplo) e assim que os dados forem enviados para o Arduino, ele deverá realizar a função imediatamente.

### 2.3.0.1 *Arduino*

"O Arduino é uma plataforma de prototipagem baseada em hardware e software flexíveis e fácil de usar"(ARDUINO, 2016). O Arduino recebe uma entrada, por exemplo, um sensor de luminosidade, ou dados via *bluetooth* ou infra vermelho, e realiza o processamento necessário para gerar uma saída. Para que essas ações aconteçam é necessário utilizar a linguagem do Arduino e o seu ambiente de desenvolvimento (ARDUINO, 2016).

O Arduino pode ser utilizado de diversas formas, como desenvolver objetos independentes conectado a um computador ou a uma rede, ou até receber e enviar dados pela internet (MCROBERTS, 2011). Neste trabalho o Arduino será utilizado para receber dados do sistema web e enviar comandos para os condicionadores de ar, luzes e projetores. A utilização do Arduino neste trabalho oferece algumas vantagens para estudantes, essas vantagens estão no Quadro 2.

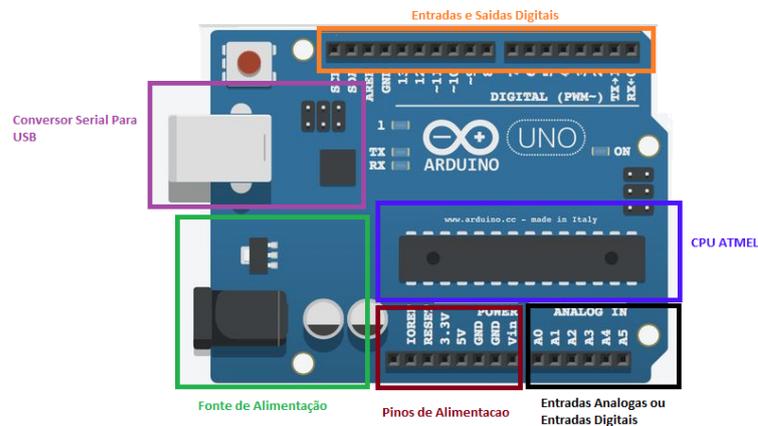
Na **Figura 3** é mostrado o modelo do Arduino Uno e alguns de seus blocos. É possível ver a CPU ATMEL, suas entradas digitais, os pinos de energia com voltagem de 3,3 e 5 volts, sua fonte de alimentação, o conversor de entrada serial para USB e suas entradas e saídas digitais.

Quadro 2 – Vantagens Arduino

Características	Conceitos
Custo	As placas de Arduino são mais baratas do que a maioria dos outros microcontroladores, pois as placas podem ser montadas à mão e isso barateia seu custo.
Multi-Plataforma	A IDE do Arduino pode ser executada em diversos sistemas operacionais enquanto outros microcontroladores são limitados a utilizar apenas o Windows.
Ambiente de programação simples	O ambiente de programação é fácil para iniciantes utilizarem e bem flexível para que usuários avançados utilizem.
Código aberto e software extensível	O software Arduino é feito em código aberto, disponível para a extensão por programadores experientes. A linguagem pode ser expandida através de bibliotecas C ++ , e as pessoas que querem entender os detalhes técnicos podem usar a linguagem de programação AVR C na qual ele se baseia.
Código aberto e hardware extensível	Os planos das placas Arduino são publicados sob uma licença Creative Commons, para que os designers de circuito experientes possam fazer a sua própria versão do módulo, estendendo-lo e melhorá-lo. Mesmo os usuários relativamente inexperientes pode construir a versão placa de ensaio do módulo, a fim de entender como ele funciona.

Fonte: Adaptado de Arduino (2016).

Figura 3 – Arduino Uno



Fonte: Adaptado de Instructables (2016)

## 2.4 Computação em Nuvem

O termo Computação em Nuvem (*Cloud Computing*) surgiu na área de telecomunicações quando seus provedores começaram a usar rede virtual privada (*virtual private network - VPN*) (KAUFMAN, 2009). Um dos principais objetivos de Computação em Nuvem é poder alcançar um melhor rendimento e distribuição de recursos e poder resolver grandes cálculos de dados (JADEJA; MODI, 2012). A Computação em Nuvem tem algumas características essenciais que são mostradas no Quadro 3.

Quadro 3 – Características da Computação Nuvem

Características	Conceitos
Serviço sobre demanda	Os serviços requisitados por clientes podem ser feitos sem a necessidade de intervenção humana.
Elasticidade sobre demanda	Os clientes podem usar o serviço na hora que quiserem e liberar na hora que quiserem e os recursos de hardware vão se adequar as necessidades do cliente.
Abstração	Os recursos são escondidos dos clientes, eles usam os recursos sem saber sua localização.
Acesso a rede	O aplicativo do cliente pode rodar em diversas plataformas utilizando uma conexão segura.
Medição de serviço	A nuvem é capaz de medir o uso de recurso para cada um dos clientes, embora os recursos sejam compartilhados entre vários clientes.
Gestão de recursos	Os recursos são atribuídos dinamicamente a partir da requisição do cliente.

Fonte: Adaptado de Dash, Mohapatra e Pattnaik (2010).

#### 2.4.1 Serviços oferecidos pela Computação em Nuvem

A seguir são apresentados quatro serviços oferecidos pela Computação em Nuvem que são Plataforma como serviço (*Platform as a service - PaaS*), Infraestrutura como serviço (*Infrastructure as a service - IaaS*), Software como serviço (*Software as a service - SaaS*) e Database como serviço (*Database as a service - DaaS*).

Na Plataforma como serviço é oferecido apoio ao cliente em todas as fases do ciclo de vida do desenvolvimento de aplicações (NEZHAD et al., 2009). A plataforma é oferecida como serviço e permite que os clientes utilizem este serviço para a construção de seus sistemas, sendo que os clientes não gastam com os recursos do serviço tais como armazenamento e poder de computação (TAO et al., 2014).

Software como serviço (*Software as a service - SaaS*): Oferece serviços de software para os clientes, mas não na forma de pacotes impossibilitando a compra (NEZHAD et al., 2009). O software é executado na Nuvem então as necessidades de instalação e execução do software não ficam no computador do cliente (TAO et al., 2014) e nenhum investimento é necessário no lado do cliente (DASH; MOHAPATRA; PATTNAIK, 2010).

Base de Dados como serviço (*Database as a service - DaaS*): Este serviço oferece banco de dados como um serviço. Ele adota uma arquitetura em que os dados de vários usuários são mantidos na mesma tabela (NEZHAD et al., 2009). Um exemplo é o serviço SSDS da Amazon que ajuda a manter a integridade dos dados e proteger os clientes de rastreamento de operações longas (WANG, 2014).

A Infraestrutura como Serviço oferece serviços de *hardware* virtual, armazenamento

e poder de computação como serviço para os clientes, fazendo assim com que os clientes aluguem os recursos em vez de ter que comprar (NEZHAD et al., 2009), (TAO et al., 2014). Algumas empresas fornecem serviços como poder de computação e armazenamento para o seus clientes de forma que eles paguem apenas o que utilizam. A Amazon EC2 (*elastic compute*) é um exemplo de um serviço nessa categoria (NEZHAD et al., 2009) (WANG, 2014).

Neste trabalho será utilizado o serviço de IaaS para hospedar o sistema web. Ele é utilizado pois o objetivo do trabalho é reduzir o consumo de energia da UFC - Campus Quixadá, logo utilizar os servidores na nuvem para hospedar o sistema web é uma solução que ajuda a diminuir os gastos no campus.

#### 2.4.2 Tipos de Nuvem

A seguir serão descritos os três tipos de nuvem que são nuvem privada, nuvem pública e híbrida.

A **nuvem privada** é gerenciada dentro de uma organização para maximizar e otimizar a utilização dos recursos existentes (DASH; MOHAPATRA; PATTNAIK, 2010). Ela tenta resolver o problema da subutilização de recursos utilizando a virtualização para reduzir os custos das cargas de trabalho (NEZHAD et al., 2009). Uma vantagem da privada é que se torna mais fácil gerenciar, manter, fazer atualizações além de ficar mais fácil realizar o controle do desenvolvimento de software (JADEJA; MODI, 2012).

As **nuvens públicas** são oferecidas e operadas por outras empresas como Amazon, Microsoft, Google e os clientes compartilham a mesma infra-estrutura, configuração e proteção de forma limitada (DASH; MOHAPATRA; PATTNAIK, 2010). Elas utilizam o conceito de pague pelo que você usa (*pay-as-you-go*) (SACHDEVA et al., 2011).

A **nuvem híbrida** é a combinação das nuvens públicas e privadas. A nuvem híbrida utiliza de outras nuvens de forma total ou parcial (NEZHAD et al., 2009). Dessa forma é mais segura a forma de controlar os dados das aplicações (JADEJA; MODI, 2012).

As nuvens públicas estão mais sujeitas a ataques. Uma forma de prevenir esses ataques é fazendo validações no lado da Nuvem como no lado do cliente (JADEJA; MODI, 2012). Apesar dessa característica este trabalho utilizará uma nuvem pública para que o sistema web seja hospedado.

### 3 TRABALHOS RELACIONADOS

Nesta seção serão apresentadas pesquisas relacionadas com este trabalho.

#### 3.1 Towards the Implementation of IoT for Environmental Condition Monitoring in Homes

Em Kelly, Suryadevara e Mukhopadhyay (2013) foi implementada uma arquitetura para o controle de uma residência, como a água, eletricidade, além de monitorar intensidade da luz, temperatura e umidade. Kelly, Suryadevara e Mukhopadhyay (2013) utilizam dos conceitos de Internet das Coisas (*internet of things - IOT*) na arquitetura para reduzir o consumo de energia fazendo com que o consumo de energia fosse realizado de acordo com a demanda.

Neste trabalho os autores Kelly, Suryadevara e Mukhopadhyay (2013) aplicam a abordagem ZigBee como padrão de comunicação na sua arquitetura e utilizam uma rede IPv6 privada para realizar a comunicação do sensor com o *gateway* servidor (KELLY; SURYADEVARA; MUKHOPADHYAY, 2013). Como o ZigBee não se comunica com o protocolo da internet, é executado um programa para que a comunicação entre o sensor que utiliza ZigBee e a rede privada que utiliza IPv6 seja realizada, com o protocolo (*User Datagram Protocol - UDP*) para enviar os pacotes ao servidor. O servidor armazena as informações em um banco de dados para que possam ser visualizadas através de um site (KELLY; SURYADEVARA; MUKHOPADHYAY, 2013). O site mostra gráficos com os valores de entradas que os sensores recebem no momento.

Kelly, Suryadevara e Mukhopadhyay (2013) serve como base para este trabalho pois discutem como é feita a comunicação entre a placa WRT54GLGateway e o servidor. Uma diferença é que este trabalho atua na área de Campus Inteligente e realizará o monitoramento e controle de luzes, condicionadores de ar e projetores, ao contrário de Kelly, Suryadevara e Mukhopadhyay (2013), que atuam na área de *Smart Homes* (casas inteligentes) e realizam o monitoramento da intensidade de luz, temperatura, umidade, etc.

Um ponto positivo neste trabalho é que serão utilizados servidores na nuvem para armazenar as informações enviadas do Arduino para o sistema web.

### 3.2 Constructing the Green Campus within the Internet of Things Architecture

No trabalho de Wang (2014) foi desenvolvida uma arquitetura que utiliza conceitos de nuvem e internet das coisas. Estes conceitos foram aplicados para criar um laboratório inteligente. A arquitetura de Wang (2014) preza por reduzir o tempo que os computadores ficam ociosos e diminuir o consumo de energia.

A arquitetura de (WANG, 2014) é dividida em três partes *hardware segment*, *middleware segment* e *presentation segment*. O *hardware segment* serve para induzir os alunos a entrarem nos laboratórios, o *middleware segment* contém um servidor que guarda todos os dados coletados e o *presentation segment* mostra informações para os estudantes e para o escritório de assuntos gerais.

Na arquitetura de Wang (2014) a placa utilizada é a ZB2530-01 que realiza a coleta de dados e envia os dados para o servidor que fica no *middleware segment*. É utilizado RFID para ler os IDs dos estudantes que entram nos laboratórios para realizar o monitoramento dos computadores que podem estar livres, ocupados ou com recursos ociosos. Todos os computadores e condicionadores de ar do laboratório tem endereço de IP, os dados dos computadores e temperatura são analisados e guardados a cada trinta minutos e caso haja um computador ocioso a placa ZB2530-01 desligará o computador.

Assim como em Wang (2014) este trabalho cria uma arquitetura para o controle e monitoramento de periféricos, na ajuda da diminuição do consumo de energia do Campus, além de mandar os dados para um servidor na nuvem para que estes sejam acessíveis a todos. As diferenças entre Wang (2014) e este trabalho é que neste trabalho não iremos realizar o controle e monitoramento de computadores. O dispositivo que Wang (2014) utiliza para realizar o monitoramento é a placa ZB2530-01. Por outro lado, neste trabalho será usado o Arduino, funcionando como um intermediário para que o sistema web se comunique com as luzes, condicionadores de ar e projetores.

### 3.3 Ubiquitous Smart Home System Using Android Application

Kumar (2014), propõe uma arquitetura para realizar o controle de dispositivos, tais como: luzes, tomadas, aquecedores, ventiladores, condicionadores de ar, sistemas de detecção de incêndio e alarmes. Sua arquitetura foi baseada na proposta de Caytiles e Park (2012) diferenciando desta última, por não utilizar um computador executando um agente para a realizar

ações. Kumar (2014) destaca que a utilização de computadores pessoais (PCs) como servidores e *web page* impactam no consumo de energia de forma negativa.

A arquitetura de (KUMAR, 2014) utiliza um *web service* rodando em um Arduino como o principal controlador para realizar as ações e sensores como interface para que a comunicação entre os periféricos e o Arduino seja realizada. O controle dos equipamentos pode ser realizado de um local remoto através de um aplicativo que foi criado para Android, esse aplicativo permite receber comandos de voz para realizar as ações. Kumar (2014) propõe no design da sua arquitetura o controle dos equipamentos e gestão de energia, por conta disso ele não utilizará PC.

Uma das diferenças entre este trabalho e o de (KUMAR, 2014) é que neste trabalho não será desenvolvido um o aplicativo Android e sim um sistema web. Outra diferença é que neste trabalho ao invés de um serviço web executando em um PC, será utilizado um servidor na nuvem.

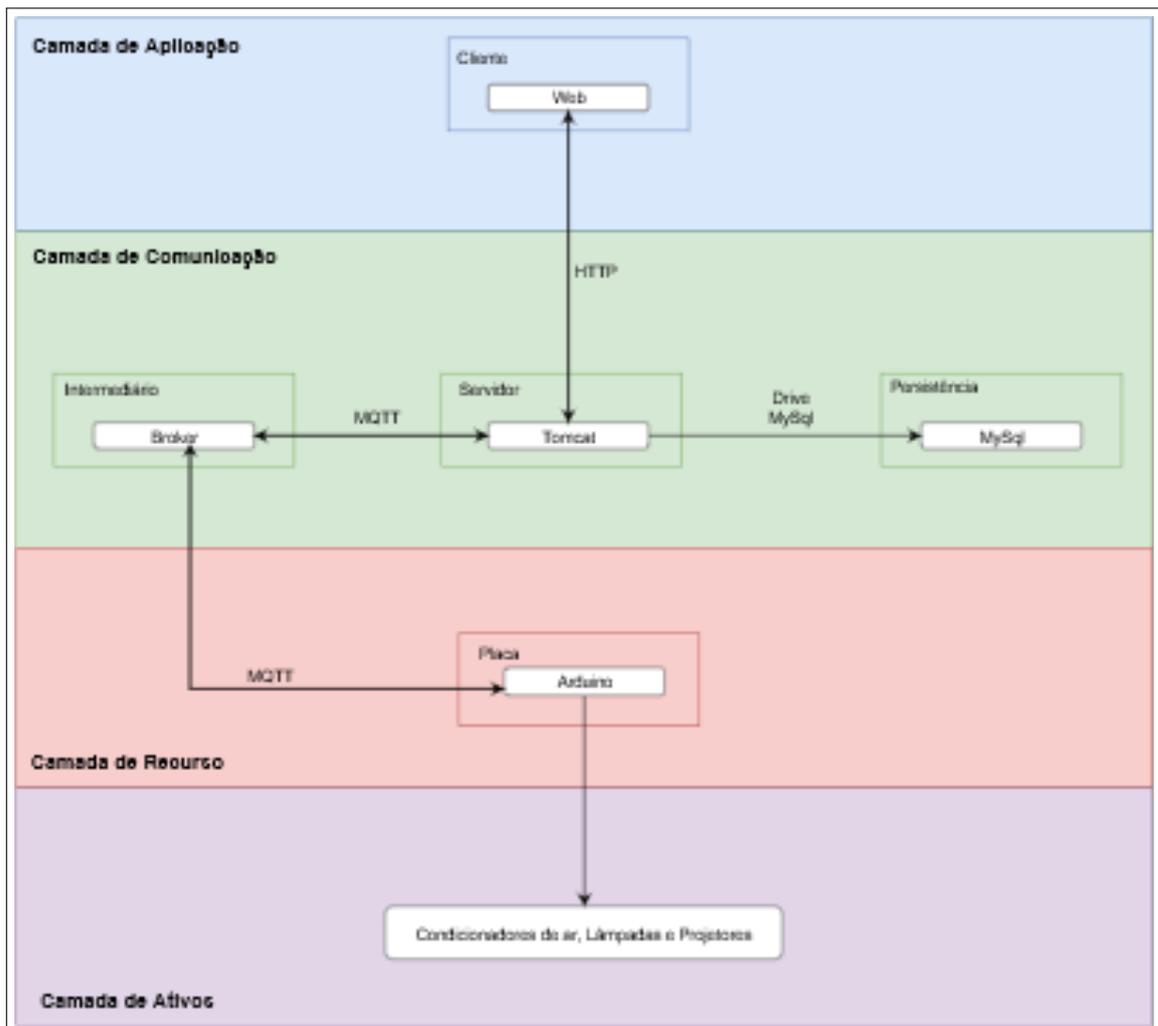
## 4 PROPOSTA

Este trabalho propõe uma ferramenta que permite controlar os projetores, lâmpadas e condicionadores de ar da UFC - Campus Quixadá baseado nos conceitos de Internet das Coisas abordados na seção **3 Fundamentação Teórica**, tendo como meta os objetivos que foram abordados na seção **1 introdução**.

### 4.1 Arquitetura do Sistema

A arquitetura do sistema está modelada de acordo com a **Figura 4** onde estão representadas as camadas de acordo com a arquitetura referencial discutida na seção **2 Fundamentação Teórica**. Os módulos do sistema são o cliente, serviço, intermediário, persistência e placa estão representados na **Figura 4** e são discutidos a seguir.

Figura 4 – Representação Arquitetura do Sistema



#### **4.1.1 Cliente**

É através do cliente que os usuários do sistema podem realizar as ações de ligar e desligar os objetos, além de aumentar e diminuir a temperatura dos condicionadores de ar. O cliente está hospedado na nuvem e se comunica diretamente com o serviço web recebendo e enviando dados utilizando o protocolo HTTP (*Hypertext Transfer Protocol*). Esse protocolo é utilizado na comunicação de vários tipos de sistemas entre eles podemos citar sistemas distribuídos e colaborativos. As ferramentas utilizadas para desenvolver o cliente foram HTML5, Thymeleaf, JavaScript e CSS. Essas ferramentas foram utilizadas pois deram agilidade no desenvolvimento do cliente.

#### **4.1.2 Serviço Web**

O serviço utiliza o servidor Apache Tomcat usando do *Framework Spring Boot* para tratar as requisições assíncronas do cliente e do intermediário (Broker) e enviando respostas a eles. O intermediário será descrito no próximo tópico. As requisições feitas pelo cliente utilizam o protocolo HTTP, já as requisições feitas pelo intermediário utilizam o protocolo MQTT.

O serviço também é responsável por persistir dados no módulo de persistência, o que será descrito nos tópicos posteriores, e enviar dados para o cliente e intermediário em um determinado intervalo de tempo.

#### **4.1.3 Intermediário**

O intermediário está situado entre o servidor e o Arduino, e sua função é enviar e receber informações referentes ao estados dos objetos para o servidor. O servidor por sua vez envia esses dados para o cliente para que essas informações sejam acessíveis aos usuários. Tanto a comunicação do intermediário com o servidor, quanto a do intermediário com o Arduino é feita utilizando o protocolo MQTT.

#### **4.1.4 Persistência**

São mantidos na persistência os dados do sistema e para isso é utilizado o sistema de gerenciamento de banco de dados (SGBD) MySQL. Ela é responsável por guardar os estados dos objetos das salas, os horários em que os objetos devem ser ligados e desligados e o local

onde cada objeto está para que seja possível realizar o controle dos objetos.

#### 4.1.5 Placa

O Arduino é o responsável por enviar informações para o intermediário sobre o estado dos objetos e de alterar o estado dos mesmos. Ele realiza as funções: aumentar e diminuir a temperatura dos condicionadores de ar, ligar e desligar o projetor e os condicionadores de ar através do infravermelho e, por fim, ligar e desligar uma lâmpada do interruptor usando um relé.

Inicialmente, o Arduino executa a função *setup*, que é responsável por se comunicar com o *broker*. Em seguida, configura a função *setCallback()* que é responsável por receber as mensagens enviadas pelo *broker* para o Arduino, além de configurar o MAC e endereço de IP do Arduino, como mostra o exemplo de código fonte 1.

A função *loop* é a responsável por verificar se o Arduino está conectado. Caso não esteja ela executa a função *reconnect*, que é mostrada no código fonte 2, para conectar o Arduino ao *broker/middleware*, essa função também é responsável por fazer com que o Arduino fique escutando as mensagens que são enviadas para os tópicos dos objetos, isso através da variável *client* e sua função *subscribe*.

Código-fonte 1 – função setup arduino

```

1 void setup(){
2   Serial.begin(9600);
3   client.setServer(server, 1883);
4   client.setCallback(callback);
5   Ethernet.begin(mac, ip);/
6
7   delay(1500);
8 }

```

Código-fonte 2 – função reconnect arduino

```

1 void reconnect() {
2   while (!client.connected()) {
3     Serial.print("Tentando se conectar...");

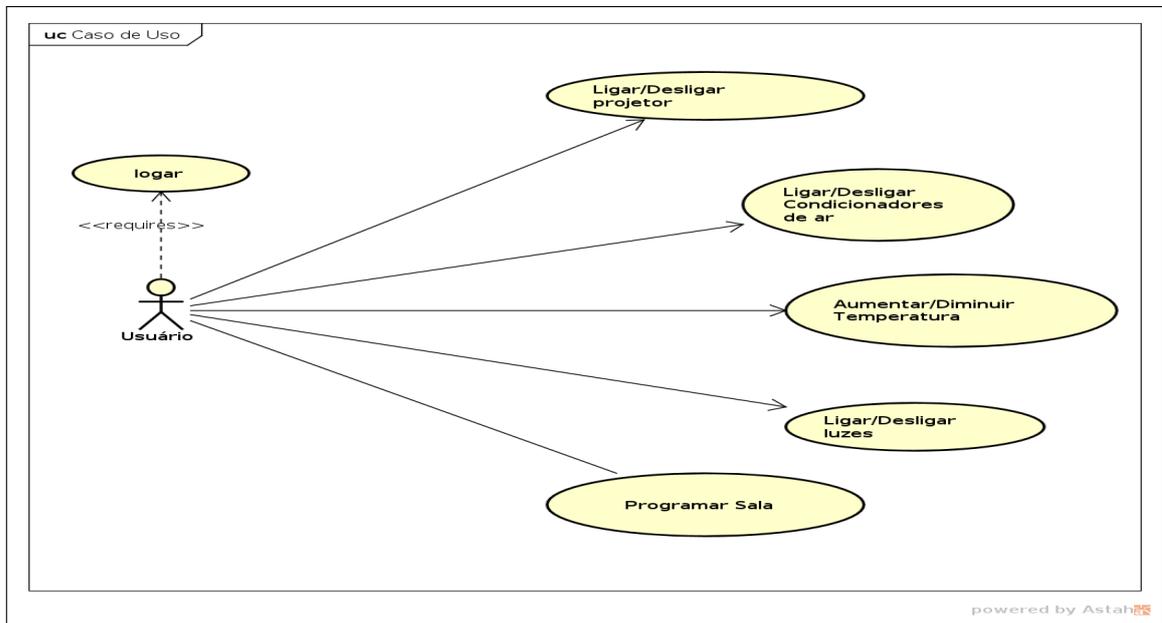
```

```
4   if (client.connect("arduinoClient")) {
5       Serial.println("connected");
6       client.subscribe("UFC/CAMPUS_QXD/Novo_Bloco/Sala_1/
7           lampada0");
8       client.subscribe("UFC/CAMPUS_QXD/Novo_Bloco/Sala_1/
9           lampada1");
10      client.subscribe("UFC/CAMPUS_QXD/Novo_Bloco/Sala_1/
11          arcondicionado0");
12      client.subscribe("UFC/CAMPUS_QXD/Novo_Bloco/Sala_1/
13          arcondicionado1");
14      client.subscribe("UFC/CAMPUS_QXD/Novo_Bloco/Sala_1/
15          projetor");
16      client.subscribe("UFC/CAMPUS_QXD/Novo_Bloco/Sala_1");
17      pinMode(lampada0, OUTPUT);
18      pinMode(lampada1, OUTPUT);
19      pinMode(lampada2, OUTPUT);
20      pinMode(lampada3, OUTPUT);
21  } else {
22      Serial.print("failed, rc=");
23      Serial.print(client.state());
24      Serial.println(" Tentar de novo em 5 s");
25      delay(5000);
26  }
```

#### 4.1.6 Visão geral das Funcionalidades

Uma visão geral sobre a interação do usuário com o sistema através do diagrama de caso de uso da **Figura 5**. Na figura podemos ver as funcionalidades que os usuários podem realizar que são: ligar/desligar um objeto, programar uma sala e logar. É importante dizer que os usuários são funcionários da UFC - Campus Quixadá.

Figura 5 – Ligar/Desligar um objeto

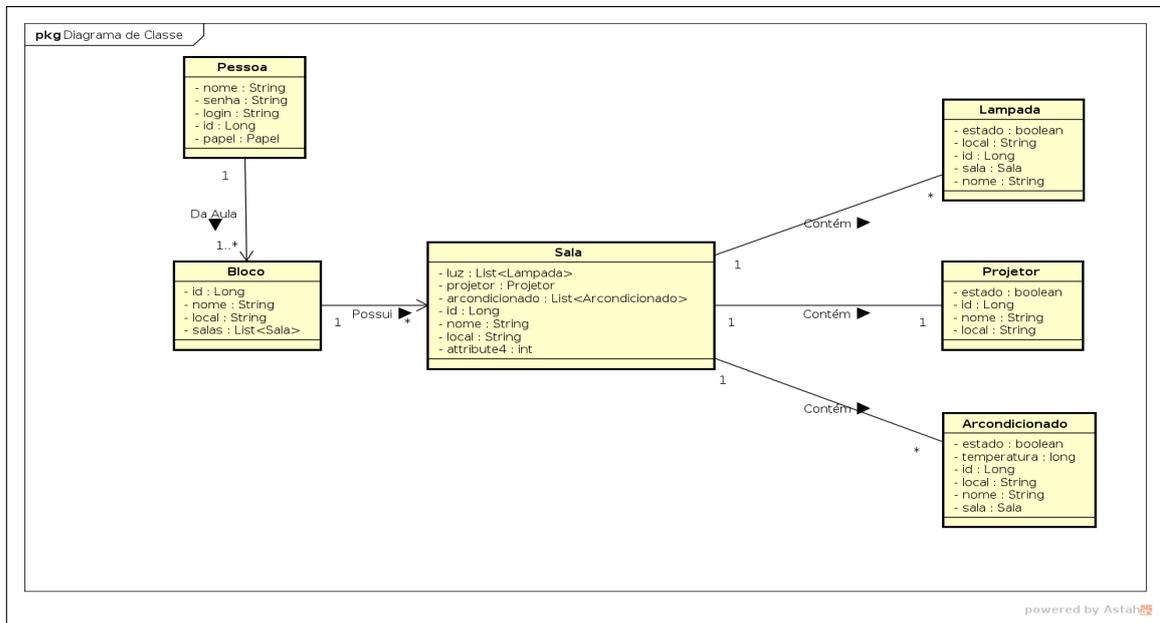


Fonte: Elaborada pelo autor

## 4.2 Funcionalidades do Sistema

A solução do sistema conta com funcionalidades que tem como objetivo automatizar alguns objetos das salas de aula e tentar diminuir o consumo de energia da UFC - Campus Quixadá. Para uma melhor compreensão das funcionalidades do sistema, podemos dividi-las em duas categorias: CRUD (*Create, Read, Update e Delete*) e controle dos objetos. As entidades do sistema estão modeladas de acordo com o diagrama da **Figura 6**. Nas próximas seções serão detalhadas cada uma das categorias.

Figura 6 – Diagrama de classe do sistema



Fonte: Elaborada pelo autor

Como podemos perceber na **Figura 6** a entidade bloco contém mais de uma sala, e essa contém um projetor e mais de uma lâmpadas e condicionadores de ar. O banco de dados está modelado de acordo com esse relacionamento.

#### 4.2.1 CRUD

Dentro dessa categoria estão as funcionalidades de cadastro, edição e remoção de um bloco e cadastro, edição e remoção das salas do bloco com seus objetos: lâmpadas, projetores e condicionadores de ar.

Quando um bloco é cadastrado, o seu atributo "*nome*" é preenchido, a partir do qual o atributo "*local*" é criado. O atributo "*local*", juntamente com um terceiro atributo, o "*id*", serve para identificar a localização dos objetos em uma sala, para permitir a troca de mensagens entre o Arduino e o servidor. A **figura 7** ilustra o passo a passo de execução do envio de dados do cliente para o controlador. Ao final da execução, os dados são persistidos no servidor de banco de dados MySQL

No cadastro de uma sala, o usuário define a quantidade de objetos que a sala tem. Neste caso, o atributo "*local*" da entidade sala é criado da seguinte forma: o caminho do pai adicionado do nome da entidade. A variável "*caminhoPai*" recebe o atributo "*local*" da entidade pai. Neste caso, a variável "*caminhoPai*" é o atributo "*local*" da entidade bloco em que a entidade

sala vai ser cadastrada, já caso dos objetos, a variável "caminhoPai" é o atributo "local" da entidade sala na qual os objetos serão cadastrados. O código fonte 3 mostra como o atributo "local" é criado.

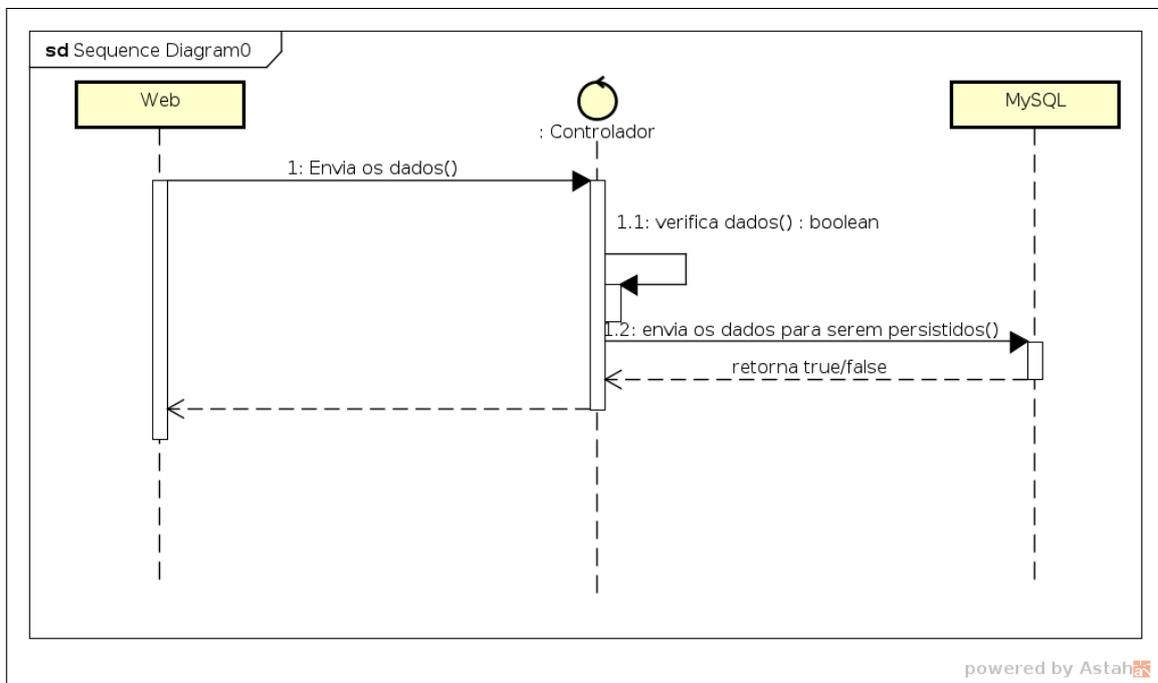
### Código-fonte 3 – Método path

```

1 public String path(String caminhoPai, String nomeObjeto){
2     StringBuilder caminhoDefinitivo = new StringBuilder();
3     caminhoDefinitivo.append(caminhoPai);
4     caminhoDefinitivo.append("/");
5     caminhoDefinitivo.append(nomeObjeto.trim().replace(" ", "_"));
6     return caminhoDefinitivo.toString();
7 }

```

Figura 7 – Diagrama de sequência exemplificando a persistência dos dados



Fonte: Elaborada pelo autor

Este diagrama exemplifica como é feita a persistência dos dados no sistema deste trabalho.

#### 4.2.2 Controle dos Objetos

O controle dos objetos é responsável pelas funcionalidades de ligar os objetos quando há aula em uma sala e de desligar os objetos quando a aula termina, ligar e desligar os objetos além de aumentar ou diminuir a temperatura dos condicionadores quando o usuário desejar.

A funcionalidade de ligar e desligar objetos pode ser realizada de forma automática desde que o usuário tenha definido os horários. Os horários são pré definidos pelo sistema.

O sistema foi pensado com horários fixos, pois as aulas sempre ocorrem nos horários 8:00 às 10:00, 10:00 às 12:00, 13:30 às 15:30, 15:30 às 17:30, 18:00 às 20:00 e 20:00 às 22:00 nos períodos de manhã, tarde e noite, podendo ou não haver aulas nas salas nesses horários. Para o caso de não ter aula na sala em um horário há a opção "sem aula". Sendo assim, quando acaba o período de aulas em uma sala ou essa sala não será utilizada durante o semestre, o usuário pode definir todos os horários dessa sala como sem aula. Nesse caso, o sistema não será responsável por ligar nem desligar os objetos dessa sala, porém pode ligar e desligar ao receber comandos diretos do usuário.

Outra forma de controlar os objetos se dá a partir do momento que o usuário está logado e deseja fazer alguma ação sobre um objeto. A principal diferença dessa funcionalidade é que além de ligar e desligar os objetos ela permite que o usuário controle a temperatura de um condicionador de ar desde que esse objeto esteja ligado.

As mensagens são enviadas do cliente para o Arduino utilizando o protocolo MQTT. A **Figura 9** mostra um diagrama de sequência e a **Figura 8** um diagrama de atividade que exemplifica como funciona o fluxo de ligar e desligar um objeto das duas funcionalidades. A mensagem é enviada do cliente para o controlador utilizando o protocolo HTTP contendo o Id da sala e o Id do objeto, o controlador verifica se a entidade está ligada ou desligada e altera o atributo estado da entidade e persiste os dados com o novo estado no banco de dados. Após ter persistido os dados, o controlador envia uma mensagem para *middleware/broker* através do protocolo MQTT, essa mensagem recebe dois parâmetros, o local onde o objeto está e a mensagem que vai ser enviada para que o objeto realize uma ação. O *middleware/broker* por sua vez manda a mensagem para o Arduino que recebe a mensagem e realiza uma ação. A seguir é mostrado o código fonte 4 que exibe como uma lâmpada é ligada e desligada.

Código-fonte 4 – Método responsável por ligar e desligar as lâmpadas

---

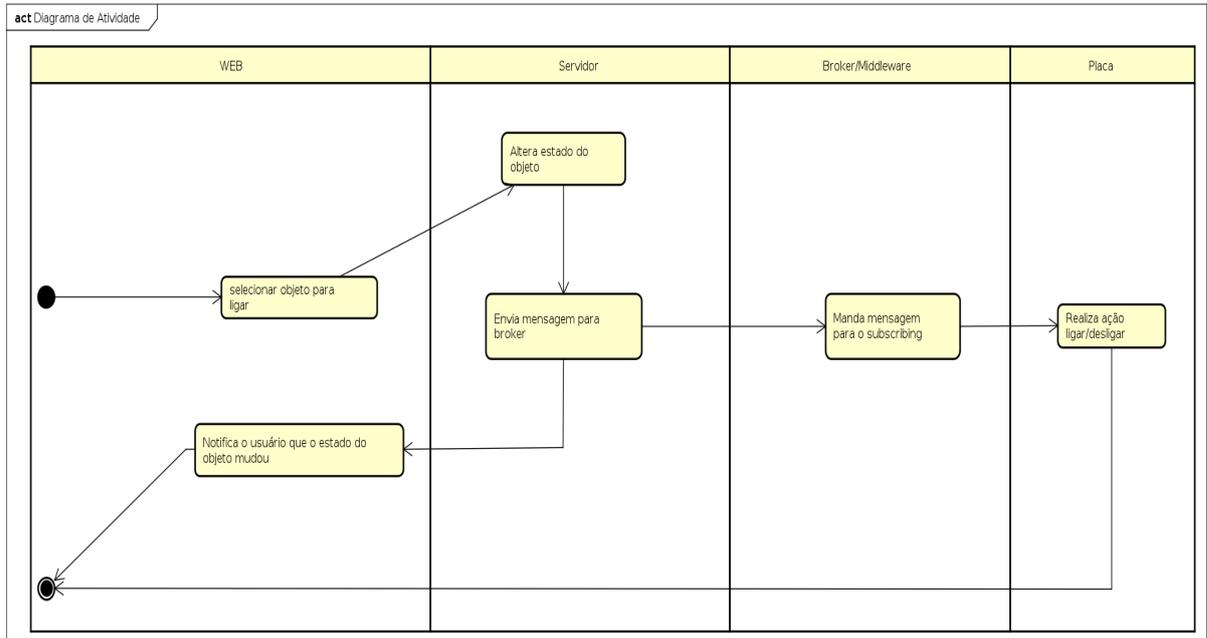
```

1 @RequestMapping(value = {"/ligar_desligar_lampada/{
    idLampada}/{idSala}"}, method = RequestMethod.GET)
2 public String ligarObjeto(@PathVariable String idLampada,
    @PathVariable String idSala, RedirectAttributes redirect
    ){
3
4     Long idLamp = Long.parseLong(idLampada);
5     Lampada lampada = lampadaService.getLampada(idLamp);
6
7     if(lampada.isEstado()){
8         lampada.setEstado(false);
9         lampadaService.saveOrEdit(lampada);
10        pubAndSub.publicarTopico(Constants.DESLIGAR, lampada.
            getLocal());
11        redirect.addFlashAttribute("desligado", mensagemService
            .getMessage(LAMPADA_DESLIGADA));
12    }else{
13        lampada.setEstado(true);
14        lampadaService.saveOrEdit(lampada);
15        pubAndSub.publicarTopico(Constants.LIGAR, lampada.
            getLocal());
16        redirect.addFlashAttribute("ligado", mensagemService.
            getMessage(LAMPADA_LIGADA));
17    }
18
19    return "redirect:/sala/detalhe_sala/"+idSala;
20 }

```

O diagrama de atividade da **Figura 8** ilustra o fluxo de como um objeto é ligado, separando por raia onde cada ação é realizada. O diagrama de sequência da **Figura 9** ilustra a troca de mensagem do caso de sucesso entre os módulos até que um objeto seja ligado ou desligado.

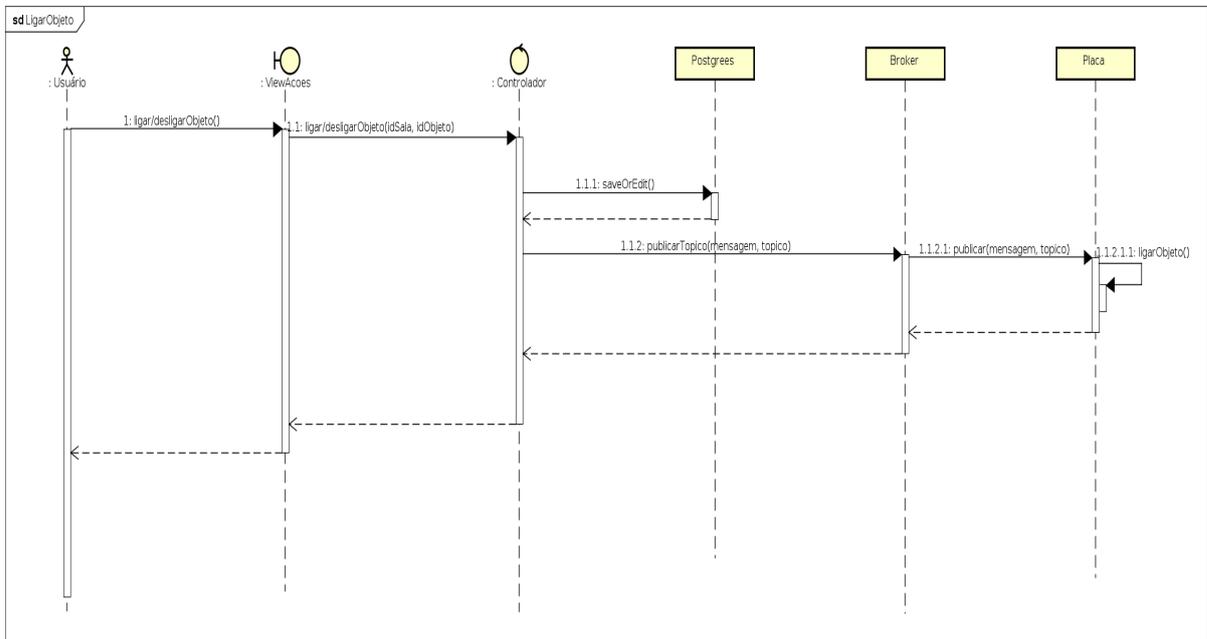
Figura 8 – Diagrama de Atividade



powered by Astah

Fonte: Elaborada pelo autor

Figura 9 – Ligar/Desligar um objeto



powered by Astah

Fonte: Elaborada pelo autor

## 5 RESULTADOS E DISCUSSÕES

As seções abaixo descrevem a avaliação e os resultados do sistema e os problemas encontrados durante a sua execução. No Apêndice A é possível ver as imagens do sistema, assim como do Arduino.

### 5.1 Avaliação e Resultados

O sistema funcionou durante a segunda feira, dia 21 de Novembro de 2016, nos blocos 1 e 2, no laboratório 4, das 13:20 as 15:30, na sala 3, das 15:50 às 17:30, e na quarta feira, dia 23 de Novembro de 2016, no bloco 1, laboratório 4, das 15:30 as 17:30. Durante esses dias foi possível observar o comportamento do sistema. É importante ressaltar que não foi possível ter acesso as lâmpadas da UFC - Campus Quixadá para controlá-las. Para contornar esse problema e verificar se o sistema está realizando a função de ligar e desligar as lâmpadas foi criado um circuito que permite controlar uma lâmpada qualquer. Esse circuito é semelhante aos circuitos das lâmpadas sendo utilizadas no campus.

No primeiro dia que o sistema funcionou foi possível ver que as funcionalidades de criar, editar e excluir bloco e sala funcionaram como esperado. As funcionalidades de ligar e desligar lâmpadas e projetores funcionaram apenas na sala 3 do bloco 2. No laboratório 4 do bloco 1 não foi possível ligar e desligar o projetor, porém a funcionalidade de ligar e desligar os condicionadores de ar apresentaram problemas em ambas as salas, esses problemas serão discutidos seção problemas.

A funcionalidade de programar os objetos de uma sala para que eles liguem e desliguem nos horários definidos apresentou problemas no laboratório 4 do bloco 1, a lâmpada e o condicionador de ar ligaram no horário definido mas apenas a lâmpada desligou no horário definido, o condicionador recebeu o comando mas não desligou, o projetor não ligou e nem desligou nos horários definidos neste laboratório. Na sala 3 do bloco 2 essa funcionalidade executou de forma correta no projetor e lâmpada ligando e desligando no horário definido, porém os condicionadores de ar não ligaram e nem desligaram.

No segundo dia o sistema foi testado apenas na sala 3 do bloco 2 e obteve fracasso na funcionalidade de ligar e desligar os condicionadores de ar e aumentar e diminuir a temperatura, além da funcionalidade de programar os objetos da sala de aula não ter sido capaz de ligar e desligar os condicionadores de ar. Todas as outras tarefas o sistema executou com êxito. Os

dados discutidos anteriormente são expostos no **Quadro 4**.

Quadro 4 – Resultados da Análise

Dados/Salas	Bloco 1/ Laboratório 4	Bloco 2/Sala 3
Horário de Início/Termino	13:30 às 17:30	13:30 às 17:30
Projektor	Não foi possível controlar o projetor.	Funcinou Corretamente.
Lâmpada	Funcionou Corretamente.	Funcionou Corretamente.
Condicionador de Ar	Foi possível ligar e mudar a temperatura, mas não foi possível desligar.	O serviço não foi capaz de realizar nenhuma ação nos condicionadores de ar dessa sala.

## 5.2 Problemas

O sistema apresentou problemas em controlar os condicionadores de ar da UFC - Campus Quixadá e controlar os projetores do bloco 1.

O problema dos condicionadores de ar ocorreu pois não foi possível mapear as funcionalidades dos condicionadores de ar do bloco 2 da UFC- Campus Quixadá. Esse problema ocorreu por que a biblioteca utilizada pelo Arduino para encontrar o código das funcionalidades de ligar, desligar, aumentar e diminuir a temperatura não foi capaz de encontrar o código (Hexadecimal) que é enviado pelo controle para o condicionador de ar. Não foi possível mapear os condicionadores de ar do térreo do bloco 1 pois as salas não estavam disponíveis, mas os condicionadores de ar dos laboratórios foram mapeados, porém na hora da execução do sistema eles apresentaram problemas na função de desligar.

Uma solução para o problema dos condicionadores de ar seria enviar o código que é enviado pelo controle sem a utilização de biblioteca, porém isso consumiria certa de 70% da memória do Arduino e não seria possível implementar o resto das funcionalidades.

O problema dos projetores do bloco 1 ocorreu pois não é possível mapear as funcionalidades dos projetores utilizando um receptor infravermelho pois eles não possuem um controle. Também não foi possível usar do Arduino para acessar os projetores através do *wireless* pois a comunicação do Arduino com o *broker* seria perdida impossibilitando que ele se comunicasse com o sistema.

### 5.3 Considerações sobre os resultados

Como podemos perceber não foi possível realizar o controle dos condicionadores de ar com êxito total, mas foi possível controlar as lâmpadas e projetores. Logo, podemos fazer um cálculo para saber se através disso conseguimos reduzir o consumo de energia.

O cálculo do consumo de energia é feito da seguinte forma:

Consumo = (potência em watt/1000) x (tempo) número de horas de funcionamento no mês = total em KWh (Quilowatt-hora)

A potência dos projetores é de 200W e as lâmpadas de 28W. Então podemos explorar alguns casos da utilização do sistema que são: primeiro o caso em que haja aula em uma sala das 8:00 até às 22:00, segundo o caso em que haja aula em uma sala das 8:00 às 10:00 e depois das 15:30 às 20:00 e o terceiro e último é o caso que haja aula das 8:00 às 12:00 e depois das 18:00 às 22:00. Iremos agora discutir o gasto de energia de utilizar o serviço proposto neste trabalho e de não utilizar este serviço nos casos citados.

No primeiro caso, se não utilizarmos o serviço proposto neste trabalho e a sala ficar com todos os objetos ligados durante esse horário o consumo de energia é de 150,08Kwh, se utilizarmos o serviço proposto neste trabalho as salas irão desligar os objetos quando um período terminar e ligar quando o outro começar, logo o consumo de energia é de 128,64Kwh. Logo podemos perceber que há um ganho no consumo de energia do primeiro caso.

No segundo caso, se não utilizarmos o serviço proposto e assumindo que no período de 15:30 às 20:00 os objetos não sejam desligados o consumo de energia é de 69,68Kwh, já utilizando o serviço proposto que desliga e liga os objetos quando um período termina o consumo de energia é de 64,32 Kwh. Embora pouco ainda há um ganho no consumo de energia.

No terceiro e último, caso se não utilizarmos o serviço proposto neste trabalho e não houve atraso para desligar os objetos o consumo de energia será o mesmo que utilizando o serviço proposto. Porém se houver um atraso de 15 minutos para desligar os objetos o consumo de energia será de 91,12Kwh, já utilizando o serviço o consumo diminui para 85,76Kwh.

Assim podemos perceber que embora este trabalho não conseguiu controlar todos os objetos propostos, ele conseguiu obter um ganho na redução de energia que é o objetivo principal do trabalho.

## 6 TRABALHOS FUTUROS

Uma possível melhoria identificada para o Arduino é fazer com que ele ficasse mais genérico possibilitando que o usuário do sistema escolhesse um Arduino e o atribuísse a uma sala fazendo assim com que o Arduino fosse responsável por fazer as configurações necessárias para possibilitar a comunicação do sistema com os objetos da sala através do *broker*.

Uma melhoria observada para o sistema seria fazer com que o login seja realizado utilizando o serviço LDAP( *Lightweight Directory Access Protocol*) que já está consolidado na UFC - Campus Quixadá. Este sistema permite uma autenticação dos usuários já cadastrados na instituição.

Outra melhoria para este trabalho seria evolui-lo para um *framework* fazendo com que as funcionalidades que foram implementadas ficassem mais genéricas de forma que elas pudessem ser utilizadas por outras aplicações que realizassem funções similares às desenvolvidas nesse trabalho.

## 7 CONSIDERAÇÕES FINAIS

Este trabalho desenvolveu um serviço capaz de controlar as lâmpadas, projetores e condicionadores de ar da Universidade Federal do Ceará - Campus Quixadá com o intuito de economizar energia e facilitar os trabalhos dos servidores da Universidade.

Esse trabalho desenvolveu uma serviço que foi capaz de controlar as lâmpadas e projetores com êxito, porém os condicionadores de ar apresentaram problemas que foram discutidos na seção **5 Resultados e Discussões**. Sendo assim este trabalho prova que é possível haver um ganho na redução do consumo de energia do campus utilizando o serviço proposto. Além de controlar a maior parte dos objetos facilitando assim o trabalho dos servidores e professores da Universidade.

## REFERÊNCIAS

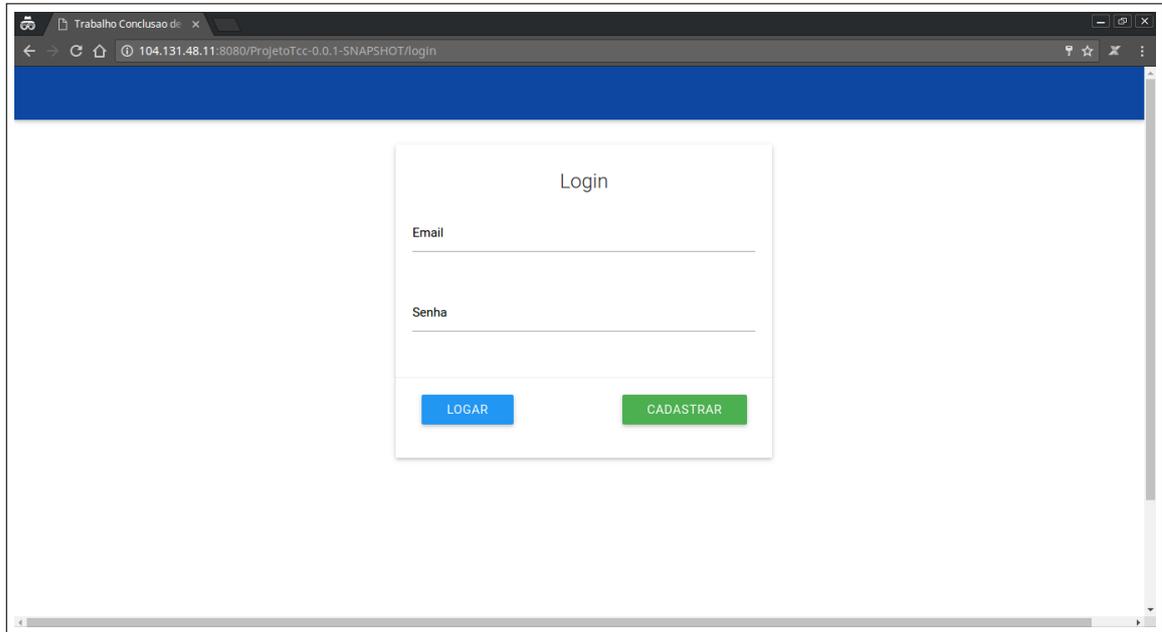
- ARDUINO. **Arduino - Introduction**. 2016. Acessado em 03 de Julho 2016. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.
- CARRO, L.; WAGNER, F. R. Sistemas computacionais embarcados. **Jornadas de atualização em informática**. Campinas: UNICAMP, 2003.
- CAYTILES, R. D.; PARK, B. Mobile ip-based architecture for smart homes. Citeseer, 2012.
- COAP. **Constrained Application Protocol**. 2016. Acessado em 15 de Novembro 2016. Disponível em: <<http://coap.technology/>>.
- CUNHA, A. F. O que são sistemas embarcados. **Saber Eletrônica**, v. 43, n. 414, p. 1–6, 2007.
- DASH, S. K.; MOHAPATRA, S.; PATTNAIK, P. K. A survey on applications of wireless sensor network using cloud computing. **International Journal of Computer science & Engineering Technologies (E-ISSN: 2044-6004)**, v. 1, n. 4, p. 50–55, 2010.
- EMBARCADOS. **MQTT - Protocolos para IoT**. 2016. Acessado em 15 de Novembro 2016. Disponível em: <<https://www.embarcados.com.br/mqtt-protocolos-para-iot/>>.
- GUBBI, J.; BUYYA, R.; MARUSIC, S.; PALANISWAMI, M. Internet of things (iot): A vision, architectural elements, and future directions. **Future Generation Computer Systems**, Elsevier, v. 29, n. 7, p. 1645–1660, 2013.
- HIRSCH, B.; NG, J. W. Education beyond the cloud: anytime-anywhere learning in a smart campus environment. In: IEEE. **Internet Technology and Secured Transactions (ICITST), 2011 International Conference for**. [S.l.], 2011. p. 718–723.
- HOLLER, J.; TSIATSIS, V.; MULLIGAN, C.; AVESAND, S.; KARNOUSKOS, S.; BOYLE, D. **From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence**. [S.l.]: Academic Press, 2014.
- INSTRUCTABLES. **Beginner Arduino**. 2016. Acessado em 03 de Julho 2016. Disponível em: <<http://www.instructables.com/id/Beginner-Arduino/>>.
- JADEJA, Y.; MODI, K. Cloud computing-concepts, architecture and challenges. In: IEEE. **Computing, Electronics and Electrical Technologies (ICCEET), 2012 International Conference on**. [S.l.], 2012. p. 877–880.
- KAUFMAN, L. M. Data security in the world of cloud computing. **IEEE Security & Privacy**, IEEE, v. 7, n. 4, p. 61–64, 2009.
- KELLY, S. D. T.; SURYADEVARA, N. K.; MUKHOPADHYAY, S. C. Towards the implementation of iot for environmental condition monitoring in homes. **Sensors Journal, IEEE**, IEEE, v. 13, n. 10, p. 3846–3853, 2013.
- KOVATSCH, M.; DUQUENNOY, S.; DUNKELS, A. A low-power coap for contiki. In: IEEE. **2011 IEEE Eighth International Conference on Mobile Ad-Hoc and Sensor Systems**. [S.l.], 2011. p. 855–860.

- KULADINITHI, K.; BERGMANN, O.; PÖTSCH, T.; BECKER, M.; GÖRG, C. Implementation of coap and its application in transport logistics. **Proc. IP+ SN, Chicago, IL, USA**, 2011.
- KUMAR, S. Ubiquitous smart home system using android application. **arXiv preprint arXiv:1402.2114**, 2014.
- LIMA, A. T. **Aplicação de Internet of Things em casas inteligentes-Serviço Aplicacional**. Tese (Doutorado) — Instituto Politécnico do Porto. Instituto Superior de Engenharia do Porto., 2014.
- MCROBERTS, M. Arduino básico. **Editora Novatec**, v. 344755160, 2011.
- MIORANDI, D.; SICARI, S.; PELLEGRINI, F. D.; CHLAMTAC, I. Internet of things: Vision, applications and research challenges. **Ad Hoc Networks**, Elsevier, v. 10, n. 7, p. 1497–1516, 2012.
- MQTT. **MQTT - Frequently Asked Questions**. 2016. Acessado em 15 de Novembro 2016. Disponível em: <<http://mqtt.org/>>.
- MQTT. **OASIS Standard**. 2016. Acessado em 15 de Novembro 2016. Disponível em: <<http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html>>.
- NEZHAD, H. R. M.; STEPHENSON, B.; SINGHAL, S.; CASTELLANOS, M. Virtual business operating environment in the cloud: conceptual architecture and challenges. In: **SPRINGER. International Conference on Conceptual Modeling**. [S.l.], 2009. p. 501–514.
- NIE, X. Constructing smart campus based on the cloud computing platform and the internet of things. In: **Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering ICCSEE**. [S.l.: s.n.], 2013. p. 1576–1578.
- PERERA, C.; ZASLAVSKY, A.; CHRISTEN, P.; GEORGAKOPOULOS, D. Context aware computing for the internet of things: A survey. **Communications Surveys & Tutorials, IEEE**, IEEE, v. 16, n. 1, p. 414–454, 2014.
- SACHDEVA, M.; RANA, P.; KAPOOR, R.; SHAHID, M. Cloud computing-pay as you go technology. In: **Proceedings of the 5th National Conference on Computing for National Development**. [S.l.: s.n.], 2011. v. 161.
- SEN.SE, M. **Mother Sen.se**. 2016. Acessado em 01 de Julho 2016. Disponível em: <<https://sen.se/mother/>>.
- SHELBY, Z.; HARTKE, K.; BORMANN, C. **The constrained application protocol (CoAP)**. [S.l.], 2014.
- TAO, F.; ZUO, Y.; XU, L. D.; ZHANG, L. Iot-based intelligent perception and access of manufacturing resource toward cloud manufacturing. **IEEE Transactions on Industrial Informatics**, IEEE, v. 10, n. 2, p. 1547–1557, 2014.
- WANG, H.-I. Constructing the green campus within the internet of things architecture. **International Journal of Distributed Sensor Networks**, Hindawi Publishing Corporation, v. 2014, 2014.

ZAFARI, F.; PAPAPANAGIOTOU, I.; CHRISTIDIS, K. Microlocation for internet-of-things-equipped smart buildings. **IEEE Internet of Things Journal**, IEEE, v. 3, n. 1, p. 96–112, 2016.

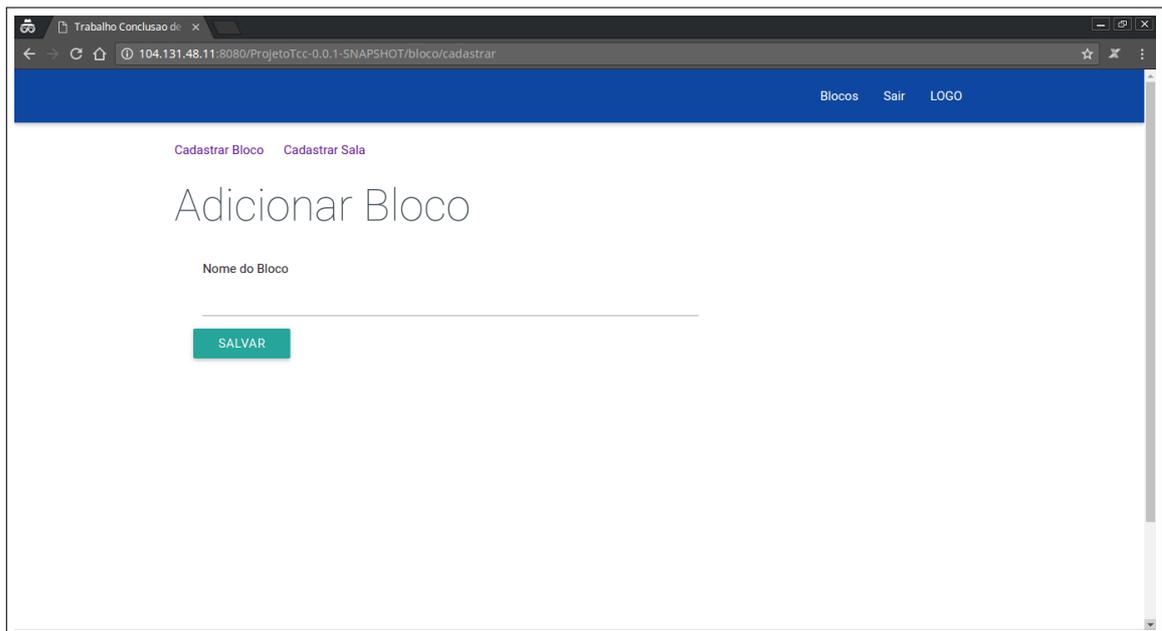
## APÊNDICE A – Imagens do Sistema

Figura 10 – Tela login



Fonte: Elaborada pelo autor

Figura 11 – Cadastro de Bloco



Fonte: Elaborada pelo autor

Figura 12 – Cadastro de Sala

Trabalho Conclusao de x

104.131.48.11:8080/ProjetoTcc-0.0.1-SNAPSHOT/sala/cadastrar

Blocos Sair LOGO

Cadastrar Bloco Cadastrar Sala

## Adicionar Sala

Nome da Sala

Número de Ar condicionados Número de Lâmpadas Selecione um bloco

SALVAR

Fonte: Elaborada pelo autor

Figura 13 – Listagem de Blocos

Trabalho Conclusao de x

104.131.48.11:8080/ProjetoTcc-0.0.1-SNAPSHOT/bloco

Blocos Sair LOGO

Cadastrar Bloco Cadastrar Sala

## Lista de blocos do campus

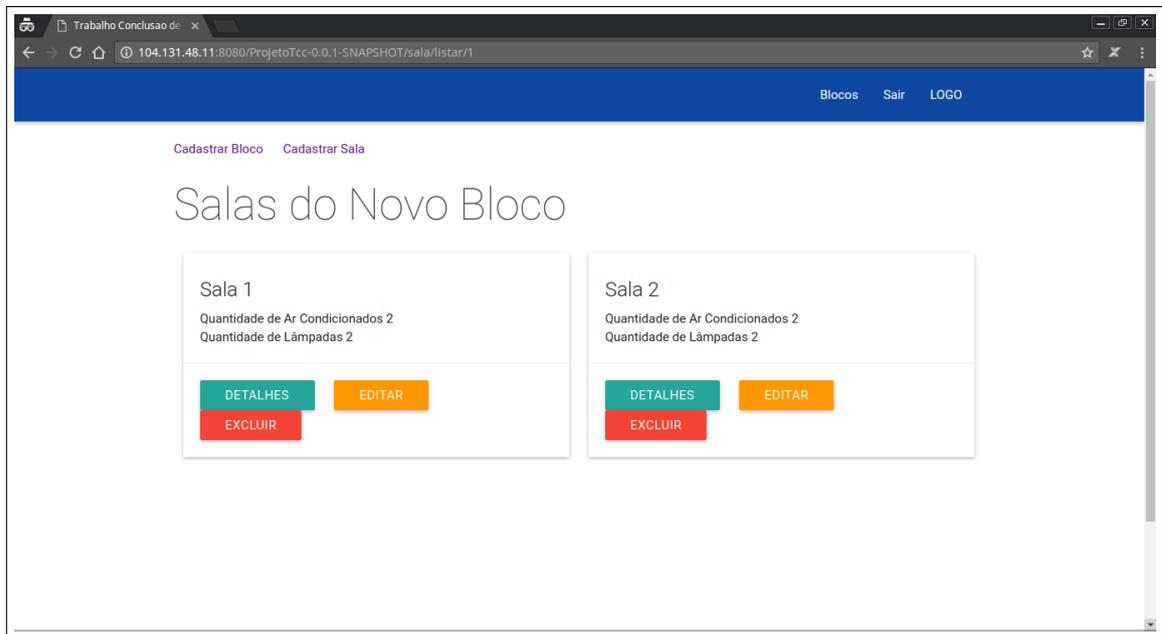
Novo Bloco

Quantidade de salas 2

VER SALAS EDITAR EXCLUIR

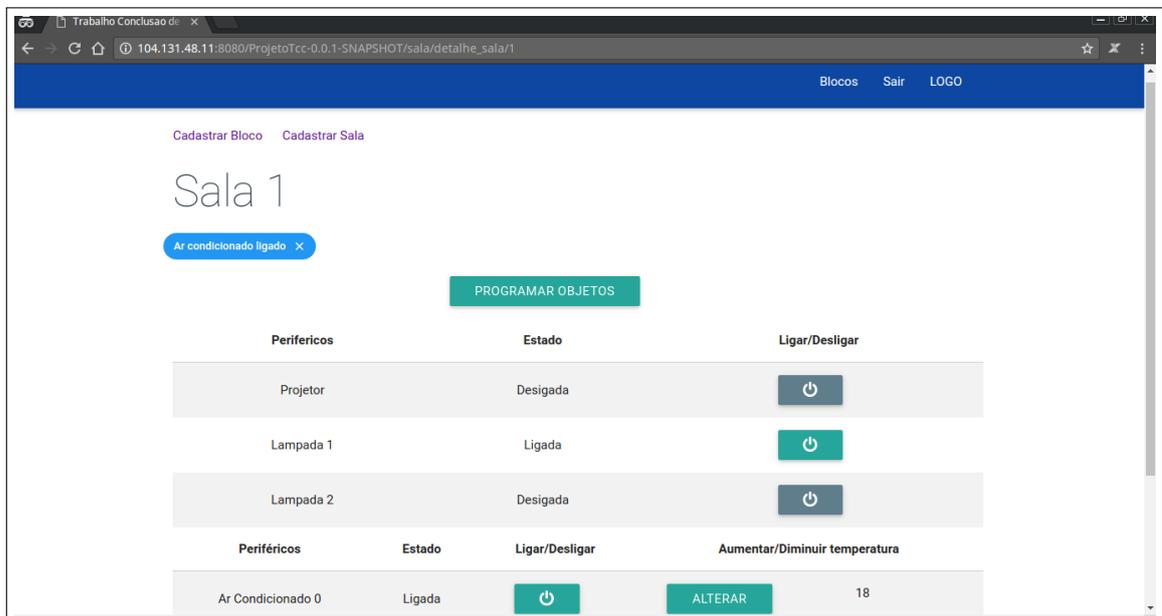
Fonte: Elaborada pelo autor

Figura 14 – Listagem de Salas



Fonte: Elaborada pelo autor

Figura 15 – Listagem de Salas



Fonte: Elaborada pelo autor

Figura 16 – Funcionalidade de Programar Objetos

Trabalho Conclusão de - x

104.131.48.11:8080/ProjetoTcc-0.0.1-SNAPSHOT/sala/editar\_objetos/1

Blocos Sair LOGO

[Cadastrar Bloco](#) [Cadastrar Sala](#)

## Novo Bloco Sala 1

### Lâmpadas e Ar Condicionados

Período Manhã	Período Tarde	Período Noite
Ligar	Ligar	Ligar
Sem aula	13:30	Sem aula
Desligar	Desligar	Desligar
Sem aula	17:30	Sem aula

SALVAR

20:00

22:00

Fonte: Elaborada pelo autor