



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS QUIXADÁ**  
**BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**FRANCISCO HUGO DUARTE GONÇALVES**

**SISTEMA PARA GERÊNCIA DE SERVIÇOS DAS SALAS DE AULA DA  
UNIVERSIDADE FEDERAL DO CEARÁ - CAMPUS QUIXADÁ UTILIZANDO  
CONCEITO DE INTERNET DAS COISAS**

**QUIXADÁ**  
**2018**

FRANCISCO HUGO DUARTE GONÇALVES

SISTEMA PARA GERÊNCIA DE SERVIÇOS DAS SALAS DE AULA DA UNIVERSIDADE  
FEDERAL DO CEARÁ - CAMPUS QUIXADÁ UTILIZANDO CONCEITO DE INTERNET  
DAS COISAS

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

Orientador: Prof. Dr. Marcio Espíndola Freire Maia

Coorientador: Prof. Felipe Alves da Costa

QUIXADÁ

2018

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)

---

G625s Gonçalves, Francisco Hugo Duarte.  
Sistema para gerência de serviços das salas de aula da Universidade Federal do Ceará - Campus Quixadá utilizando conceito de Internet das Coisas / Francisco Hugo Duarte Gonçalves. – 2018.  
52 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2018.  
Orientação: Prof. Dr. Marcio Espíndola Freire Maia.  
Coorientação: Prof. Felipe Alves da Costa.

1. Internet das Coisas. 2. Sistemas Embarcados (Computadores). 3. Bluetooth. I. Título.

CDD 005

---

FRANCISCO HUGO DUARTE GONÇALVES

SISTEMA PARA GERÊNCIA DE SERVIÇOS DAS SALAS DE AULA DA UNIVERSIDADE  
FEDERAL DO CEARÁ - CAMPUS QUIXADÁ UTILIZANDO CONCEITO DE INTERNET  
DAS COISAS

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

Aprovado em: \_\_/\_\_/\_\_\_\_.

BANCA EXAMINADORA

---

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

---

Prof. Felipe Alves da Costa (Coorientador)  
Universidade Estadual do Ceará - UECE

---

Prof. Dr. Paulo Antonio Leal Rego  
Universidade Federal do Ceará (UFC)

---

Prof. Me. Marcos Dantas Ortiz  
Universidade Federal do Ceará (UFC)

A Deus que nunca me deixou faltar nada apesar de todas as dificuldades encontradas pelo caminho. À minha família, amigos e namorada, por suas dedicatórias e ajuda prestada nos momentos felizes e tristes, principalmente a minha mãe que me apoiou sempre.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por todas as oportunidades e por ser minha fonte de força em todos os momentos.

Agradeço a minha família em especial a minha mãe, por todo apoio e carinho que me fizeram seguir em frente apesar de todas as dificuldades.

Agradeço a minha namorada, Gabrielly Soares, pelo apoio, companheirismo e por estar ao meu lado me incentivando a melhorar sempre.

Agradeço ao meu orientador, Márcio Maia e ao meu coorientador, Felipe Alves, por toda paciência e ajuda prestada por ambos durante todo desenvolvimento deste trabalho.

Agradeço a todos meus professores, todos que passam na minha vida, sem exceções, pois vocês me proporcionaram o melhor conhecimento possível, além de me auxiliar no processo de formação pessoal e profissional.

Agradeço a todos meus amigos, pelos incríveis momentos vividos, por dividir as dificuldades e multiplicar os conhecimentos adquiridos. Em especial, gostaria de agradecer ao Silvino Deolino Neto e Emmanuel Cainã Alves Melo, vocês foram fundamentais para o sucesso deste trabalho.

“Talvez não tenha conseguido fazer o melhor,  
mas lutei para que o melhor fosse feito. Não sou  
o que deveria ser, mas Graças a Deus, não sou o  
que era antes.”

(Marthin Luther King)

## RESUMO

Com o rápido aumento no número de dispositivos conectados a internet, o conceito de *Internet of Things* vem sendo cada vez mais estudado e utilizado para solucionar problemas do cotidiano da população mundial. Seguindo a ideia de utilizar o conceito de IoT para facilitar a vida das pessoas, esse trabalho propõe a criação de um sistema capaz de gerenciar os serviços das salas de aula da Universidade Federal do Ceará - Campus Quixadá. Mas precisamente o sistema desenvolvido neste trabalho gerencia o uso das lâmpadas das salas de aula e a presença dos alunos do campus, utilizando a tecnologia *Beacon*. Desta forma o sistema utiliza o conceito de *Internet of Things* para auxiliar os professores, alunos e o campus em geral.

**Palavras-chave:** Internet das Coisas. Sistemas Embarcados. Bluetooth.

## **ABSTRACT**

With the increase number of devices connected to the internet, the concept of Internet of Things has been increasingly studied and used to solve day to day problems of the world population. Following the idea of use the IoT concept to make people's life easier, this paper proposes the creation of a system capable of manage the classroom resources of the Federal University of Ceara - Quixadá Campus. More precisely the system developed in this paper manages the use of classroom lights and the presence of students in the campus, using the Beacon technology. This way the system uses the Internet of Things concept to help professors, students and the campus as a whole.

**Keywords:** Internet of Things. Embedded Systems. Bluetooth.

## LISTA DE FIGURAS

Figura 1 – Arquitetura Referencial . . . . .	18
Figura 2 – Arduino UNO . . . . .	23
Figura 3 – Arquitetura do sistema . . . . .	32
Figura 4 – Tela de detalhes da sala . . . . .	33
Figura 5 – Tela de sala próxima . . . . .	34
Figura 6 – Módulos <i>Bluetooth</i> . . . . .	43
Figura 7 – Tela de reserva de uma sala . . . . .	44

## LISTA DE TABELAS

Tabela 1 – Comparação entre os trabalhos relacionados . . . . .	30
Tabela 2 – Ações executadas de forma automática pelo sistema . . . . .	38
Tabela 3 – Possíveis status de um <i>Beacon</i> . . . . .	41
Tabela 4 – Especificações dos módulos <i>Bluetooth</i> . . . . .	42
Tabela 5 – Especificações dos dispositivos . . . . .	46

## LISTA DE CÓDIGOS-FONTE

Código-fonte 1	–	Obtendo ID da sala através da mensagem do <i>Beacon</i> . . . . .	35
Código-fonte 2	–	Publicando uma mensagem usando Paho Java Client . . . . .	36
Código-fonte 3	–	Requisição para desligar as luzes de uma sala . . . . .	36
Código-fonte 4	–	Arduíno se inscrevendo em tópicos . . . . .	39
Código-fonte 5	–	Função executada ao receber uma mensagem do interceptador . . . .	39
Código-fonte 6	–	Método para realizar presença de um usuário . . . . .	44

## LISTA DE ABREVIATURAS E SIGLAS

IoT	<i>Internet of Things</i>
UFC	Universidade Federal do Ceará
RFID	<i>Radio Frequency Identification</i>
M2M	<i>Machine To Machine</i>
LTE	<i>Long Term Evolution</i>
BLE	<i>Bluetooth Low Energy</i>
API	<i>Application Programming Interface</i>
JSON	<i>JavaScript Object Notation</i>
HTTP	<i>Hypertext Transfer Protocol</i>
UUID	<i>Universally Unique Identifier</i>
MQTT	<i>Message Queue Telemetry Transport</i>
REST	<i>Representational State Transfer</i>

## 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</b>	<b>16</b>
<b>2.1.1</b>	<i>Arquitetura IoT</i>	<b>17</b>
<b>2.2</b>	<b>Ambientes Inteligentes</b>	<b>19</b>
<b>2.2.1</b>	<i>Automação Residencial</i>	<b>19</b>
<b>2.2.2</b>	<i>Cidades Inteligentes</i>	<b>20</b>
<b>2.2.3</b>	<i>Campi Inteligentes</i>	<b>20</b>
<b>2.3</b>	<b>Tecnologias de Suporte</b>	<b>21</b>
<b>2.3.1</b>	<i>Hardware</i>	<b>21</b>
<b>2.3.2</b>	<i>Comunicação</i>	<b>22</b>
<b>2.3.3</b>	<i>Middleware</i>	<b>24</b>
<b>3</b>	<b>OBJETIVOS</b>	<b>26</b>
<b>3.1</b>	<b>Objetivo Geral</b>	<b>26</b>
<b>3.2</b>	<b>Objetivos específicos</b>	<b>26</b>
<b>4</b>	<b>TRABALHOS RELACIONADOS</b>	<b>27</b>
<b>4.1</b>	<b>GreatRoom: Uma Aplicação Android Baseada em Proximidade para a Criação de Salas Virtuais Inteligentes</b>	<b>27</b>
<b>4.2</b>	<b>Sistema inteligente de controle e monitoramento de ambiente de laboratórios de análises químicas</b>	<b>28</b>
<b>4.3</b>	<b>Desenvolvimento de um sistema de automação residencial de baixo custo aliado ao conceito de Internet das Coisas (IoT)</b>	<b>29</b>
<b>5</b>	<b>PROPOSTA</b>	<b>31</b>
<b>5.1</b>	<b>Arquitetura do Sistema</b>	<b>31</b>
<b>5.1.1</b>	<i>Cliente</i>	<b>31</b>
<b>5.1.1.1</b>	<i>Web</i>	<b>31</b>
<b>5.1.1.2</b>	<i>Mobile</i>	<b>33</b>
<b>5.1.2</b>	<i>Servidor</i>	<b>35</b>
<b>5.1.2.1</b>	<i>Serviço Web</i>	<b>35</b>
<b>5.1.2.2</b>	<i>API</i>	<b>38</b>
<b>5.1.3</b>	<i>Interceptor</i>	<b>38</b>

5.1.4	<i>Placa</i> . . . . .	39
6	<b>RESULTADOS E DISCUSSÕES</b> . . . . .	42
6.1	<i>Avaliação</i> . . . . .	45
7	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	47
	<b>REFERÊNCIAS</b> . . . . .	48

## 1 INTRODUÇÃO

Internet das Coisas, mais conhecida como *Internet of Things* (IoT), segundo Minerva, Biru e Rotondi (2015) é o conceito de uma rede que conecta objetos físicos (Coisas) e possibilita a troca de informações entre eles. Esses objetos são programáveis e possuem capacidade de sensoriamento e atuação, possibilitando a coleta de informações sobre os mesmos ou sobre o ambiente em que eles atuam. Minerva, Biru e Rotondi (2015) ainda falam que essas informações podem ser enviadas para outro objeto, com isso, ações pré-programadas podem ser executadas. Por exemplo, é possível que com base em informações coletadas, o estado em que um objeto se encontra seja alterado a qualquer momento, por qualquer outro objeto.

A IoT é destaque no cenário global, proporcionando o desenvolvimento de uma grande quantidade de aplicações e amplificando as chances de criar inovações em produtos e serviços (SANTOS et al., 2015). Segundo Atzori, Iera e Morabito (2010), essas aplicações podem ser agrupadas em domínios como: transporte e logística, saúde, pessoal, área social e ambiente inteligente. O domínio *Smart Environment* (Ambiente Inteligente), é um pequeno mundo onde diversos dispositivos são utilizados com o intuito de tornar a vida dos habitantes daquele ambiente mais confortável (COOK; DAS, 2004). Segundo o conceito de IoT dos autores Minerva, Biru e Rotondi (2015) citado acima, podemos perceber que a IoT nos proporciona o gerenciamento de serviços de um determinado ambiente através de dispositivos inteligentes. Essa capacidade de gerenciamento torna ambientes comuns em *Smart Environment*, onde é possível, por exemplo, controlar o uso de recursos como lâmpadas e condicionadores de ar (NIE, 2013).

Dentro do domínio *Smart Environment*, temos o *Smart Campus* (Campus Inteligente), em que a IoT é aplicada em um ambiente educacional e que segundo Hirsch e Ng (2011), possui algumas características que garantem o bom funcionamento de um *Smart Campus*, estas características auxiliam desde a sustentabilidade, controle de gastos até o gerenciamento da organização do campus, como a gestão de processos. Em relação as características citadas por Hirsch e Ng (2011), vale ressaltar o uso da IoT como ferramenta para potencializar a sustentabilidade de um *Smart Campus*, a mesma também é intitulada pelos autores como *iGreen*.

Utilizando este contexto, o presente trabalho propõe a construção de um sistema para a gerencia de serviços das salas de aula da Universidade Federal do Ceará (UFC) - Campus Quixadá. O sistema será capaz de controlar as lâmpadas das salas de aula, reduzindo o desperdício de energia proveniente do tempo ocioso em que as mesmas ficam ligadas. O sistema desenvolvido neste trabalho também será capaz de controlar o serviço de presença dos alunos

em aulas cadastradas no mesmo. Tendo como base os conceitos de IoT e *Smart Campus*, o desenvolvimento do sistema proposto visa contribuir com os professores, alunos e o Campus em geral, devido a gerência do serviço de presença e a redução do desperdício de recursos como a energia, provinda do controle das lâmpadas.

Além da seção de introdução, este trabalho está dividido da seguinte maneira: a seção 2 de fundamentação teórica, mostra os principais conceitos que são importantes para o entendimento deste trabalho. Em seguida a seção 3 tem como função deixar claro os objetivos gerais e específicos que esse trabalho almeja alcançar. A seção 4, mostra os trabalhos relacionados a este, bem como suas semelhanças e diferenças. A seção 5 mostra o processo de desenvolvimento dos módulos do sistema. Na seção 6 são apresentados os resultados obtidos no decorrer do desenvolvimento deste projeto, bem como a validação destes resultados. Finalmente a seção 7, onde são feitas as considerações finais do trabalho e as sugestões de trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 Internet das Coisas

Internet das Coisas, popularmente conhecida como *Internet of Things*, ou apenas IoT, é um conceito que vem sendo destaque no cenário tecnológico. O termo foi citado pela primeira vez por Ashton (2009), cofundador e diretor executivo da Auto-ID Center<sup>1</sup>. Ashton (2009) afirma que a maioria dos computadores dependem da interação humana para obter informações, entretanto os seres humanos não são bons em coletar dados. Com o advento da tecnologia de sensores e *Radio Frequency Identification (RFID)*<sup>2</sup>, foi possível fazer com que máquinas realizassem essa tarefa sem as limitações humanas. Ainda segundo Ashton (2009) a Internet das Coisas tem potencial para mudar o mundo assim como a internet o fez.

A Internet das Coisas proporciona a criação de novos serviços e outras abordagens para os serviços já existentes. Prédios inteligentes, *e-health*, salas de aula inteligentes e aprendizagem melhorada são exemplos de possíveis cenários onde a aplicação da IoT desempenha um papel fundamental (ATZORI; IERA; MORABITO, 2010) e (MIORANDI et al., 2012). Segundo Atzori, Iera e Morabito (2010) a ideia base da IoT é a presença de uma grande quantidade de objetos inteligentes como sensores e atuadores, onde esses objetos são capazes de interagir e de cooperar para atingir objetivos comuns. Acrescentando essa ideia, os autores Minerva, Biru e Rotondi (2015) definem Internet das Coisas como uma rede que conecta objetos e possibilita a troca de informações entre eles. Os autores acrescentam que esses objetos são programáveis e possuem capacidade de sensoriamento e atuação, dessa maneira é possível que esses objetos coletem informações sobre o ambiente em que estão inseridos e a partir desses dados coletados, ações pré-definidas possam ser executadas.

O fato da Internet das Coisas estar cada vez mais presente no dia-a-dia das pessoas, e do grande número de aplicações que ela torna possível, popularizou este conceito. Como um exemplo dessa popularização Lima (2014) mostra que, em 2003 o número de dispositivos conectados à internet era cerca de 500 milhões e a população mundial na mesma época era 6.3 bilhões. Em 2012 o número de dispositivos ultrapassou de forma assustadora a população humana, onde existiam mais de 8.7 bilhões de dispositivos para cerca de 7 bilhões de pessoas. Segundo Evans (2012) em 2020, teremos uma média de 50 bilhões de dispositivos conectados à

<sup>1</sup> <<https://autoidlabs.org/>>

<sup>2</sup> RFID é uma tecnologia de rádio de curto alcance usada para mapear e transmitir informações digitais. (LANDT, 2005)

internet.

Atzori, Iera e Morabito (2010) agrupam a Internet das Coisas em quatro domínios: transporte e logística, saúde, ambiente inteligente, pessoal e social. O **domínio de transporte e logística** propõe melhorias nos meios de transporte em geral; através do uso de sensores e atuadores é possível enviar informações importantes para por exemplo controlar o tráfego ou monitorar mercadorias que estão sendo transportadas. No **domínio da saúde**, os benefícios trazidos pelo uso da IoT podem ser agrupadas em: rastreamento de objetos e pessoas, identificação e autenticação, coleta automática de dados e detecção. No **domínio pessoal e social**, encaixam-se as aplicações que permitem ao usuário interagir com outras pessoas; por exemplo, as "coisas" podem enviar mensagens automaticamente para os amigos do usuário, com a finalidade de informar sobre viagens, encontrar pessoas com os mesmos interesses ou até mesmo marcar jogos de futebol. O domínio de **ambiente inteligente** faz com que o uso de sensores e atuadores distribuídos em um determinado local, como casas ou escritórios, possam tornar a vida dos habitantes mais confortável; como as lâmpadas de uma casa, que aumentam sua intensidade de acordo com o horário.

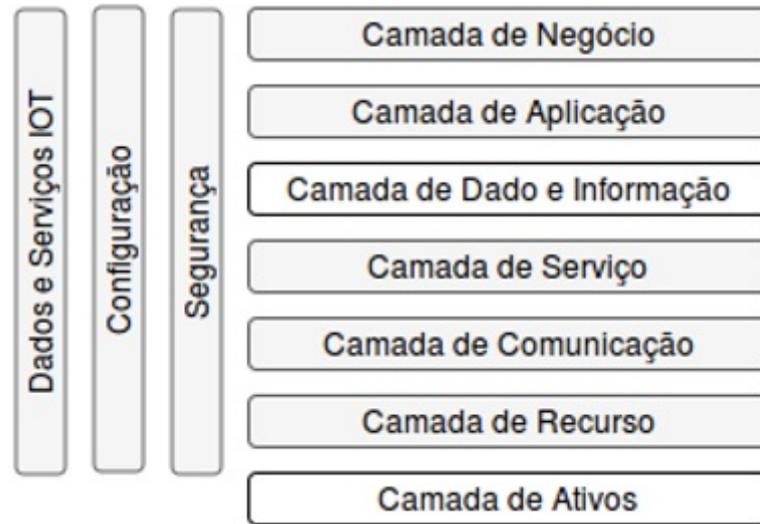
Este trabalho está inserido no domínio de ambiente inteligente e baseia-se na definição dos autores Minerva, Biru e Rotondi (2015) sobre Internet das Coisas, pois nesta definição é possível entender mais claramente o poder que os objetos inteligentes tem, onde os mesmos podem coletar informações do ambiente em que estão inseridos e através dos dados coletados podem modificar o estado de um outro objeto. Por exemplo, com base na temperatura de um comodo, é possível desligar ou ligar o condicionador de ar.

### **2.1.1 Arquitetura IoT**

Uma arquitetura mostra os elementos conceituais e concretos que compõe a Internet das Coisas, bem como as relações entre esses elementos. O desenho de uma arquitetura depende do seu princípio. A aplicação pode ter como foco, por exemplo, garantir confiança da aplicação, segurança ou voltado ao reuso em diferentes domínios. A Figura 1 mostra uma arquitetura referencial da IoT, a mesma possui maiores detalhes sobre os elementos que a compõe.

A arquitetura apresentada na Figura 1 que foi adaptada de Holler et al. (2014), descreve as camadas e quais elementos as compõem. As camadas não se comunicam exclusivamente com as camadas adjacentes, nem é obrigatório o uso de todas as camadas, sendo viável, dependendo do caso, a implementação de uma camada dentro da outra. Segundo Holler

Figura 1 – Arquitetura Referencial



Fonte – Adaptação de Holler et al. (2014).

et al. (2014) a **camada de ativos** é composta por todos elementos do ambiente que serão sensoreados, monitorados ou controlados pela aplicação, como por exemplo, condicionadores de ar e lâmpadas. Na **camada de recursos** encontra-se os dispositivos que irão monitorar, controlar ou sensorear os elementos da camada de ativos, como sensores e atuadores. A **camada de comunicação** é responsável por realizar a comunicação da camada de serviço com a camada de recursos, por exemplo, é essa camada que com base na lógica da aplicação informa a um atuador o que ele deve fazer em determinada situação. A **camada de serviço** é responsável por hospedar os serviços da aplicação, nela está contido os servidores. **Camada de dado e informação** é a camada responsável por controlar a troca de mensagens e o fluxo de informações da aplicação. A **camada de aplicação** possibilita a comunicação do usuário com os dispositivos da aplicação, como por exemplo, página web onde o usuário pode cadastrar informações importantes para controlar os serviços de uma sala de aula. A **camada de negócio** possui toda a lógica de negócio das partes interessadas, como o sistema deve agir e o que ele deve fazer. As camadas de **segurança**, **configuração** e **dados e serviços IoT** são camadas implementadas dentro das demais camadas da aplicação; elas são responsáveis por garantir características como segurança e manutenção do sistema.

A arquitetura referencial de Holler et al. (2014) servirá como base para o desenvolvimento do sistema proposto neste trabalho. Ela auxiliará na divisão correta dos recursos utilizados no sistema entre as camadas da arquitetura referencial, de modo a facilitar e deixar mais clara as responsabilidades de cada recurso.

## 2.2 Ambientes Inteligentes

Ambientes Inteligentes são ambientes eletrônicos sensíveis e adaptáveis que respondem às ações de pessoas e objetos para atenderem as suas necessidades (AARTS; WICHERT, 2009). Segundo Aarts e Wichert (2009), o conceito de Ambiente Inteligente inclui todos objetos físicos presentes e os associa à interação humana, trazendo assim uma maior eficiência, criatividade e bem-estar pessoal. Nie (2013) define Ambientes Inteligentes como ambientes onde existe troca de dados entre dispositivos, possibilitando o acesso e controle de objetos que estejam inseridos neste ambiente como lâmpadas e condicionadores de ar. Como é possível perceber, as definições de Ambiente Inteligente não divergem, os autores Cook e Das (2004) afirmam que um Ambiente Inteligente é um pequeno mundo onde dispositivos são usados para deixar a vida de seus habitantes mais confortável. Ambientes Inteligentes é um dos domínios da Internet das Coisas (ATZORI; IERA; MORABITO, 2010). A aplicação da Internet das Coisas em diferentes ambientes, com diferentes finalidades cria uma série de possibilidades como automação residencial, campus inteligentes e cidades inteligentes. A seguir dissertaremos mais sobre o uso da Internet das Coisas nesses diferentes ambientes.

### 2.2.1 Automação Residencial

Aarts e Wichert (2009) previram que apartamentos do futuro seriam equipados com sensores de movimento, sistemas de iluminação inteligente, entre outras tecnologias que facilitariam a vida dos habitantes desses apartamentos. Essa previsão se cumpriu e hoje temos diversos tipos de equipamentos que facilitam o dia-a-dia de pessoas em suas residências. Segundo Holler et al. (2014), com a comunicação *Machine-To-Machine* (M2M) e o advento da Internet das Coisas, objetos como televisores e máquinas de lavar estão tornando-se objetos inteligentes que podem ser monitorados e controlados através de interfaces de rede.

Wanzeler, Fülber e Merlin (2016), por exemplo, utilizam o conceito de Internet das Coisas para realizar o controle da iluminação e temperatura de uma residência. Entretanto, a Automação Residencial vai muito além de controlar luzes e temperatura de determinados cômodos. Podemos, por exemplo, colocar sensores de fumaça na residência, dessa forma o sistema consegue identificar caso ocorra algum incêndio e notificar o incidente aos bombeiros para que o fogo seja apagado o mais rápido possível. É possível automatizar sistemas de segurança, para que, através de sensores de movimento, o sistema notifica ao usuário que existe

movimentos estranhos em sua casa, facilitando assim o acionamento da polícia.

### 2.2.2 *Cidades Inteligentes*

O conceito de Cidade Inteligente, ou *Smart City*, é definido por Zanella et al. (2014) como o uso da IoT para auxiliar assuntos públicos. O autor diz que o objetivo é aumentar o número de serviços oferecidos à população, aumentar a qualidade desses serviços e diminuir o seu custo. A aplicação da Internet das Coisas nos ambientes urbanos pode trazer diversos benefícios como a melhoria nos serviços de transporte, onde a IoT pode ser usada para informar aos usuários a localização dos transportes públicos e o tempo que demoram para chegar em um determinado ponto. Outros possíveis benefícios são a coleta de lixo e segurança, além da grande quantidade de dados que os dispositivos coletam deste ambiente, possibilitando utilizar esses dados para aumentar a transparência e promover ações de melhorias no ambiente urbano.

Zanella et al. (2014) fornecem exemplos de aplicações de *Smart Cities*. Em muitas cidades, existem leis para reduzir a poluição sonora em determinados horários do dia e a IoT pode ser uma solução para auxiliar a monitorar a poluição sonora em um local através de sensores que detectam esses ruídos. Também é possível construir um mapa espaço-temporal da poluição sonora nas áreas que utilizarem esse serviço. Esses sensores podem ainda serem usados na segurança, utilizando algoritmos para diferenciar sons que indiquem algum crime como barulho de brigas ou vidros quebrando (ZANELLA et al., 2014).

### 2.2.3 *Campi Inteligentes*

Segundo Nie (2013), o avanço da Internet das Coisas e da Computação em Nuvem proporcionou o surgimento dos Campi Inteligentes ou *Smart Campus*. Os autores Hirsch e Ng (2011) dividem os Campi Inteligentes em pilares relacionados ao bom funcionamento dos Campi Inteligentes, são eles: *iLearning*, *iSocial*, *iGreen*, *iHealth*, *iManagement* e *iGovernance*.

Hirsch e Ng (2011) dão uma breve descrição sobre cada um dos pilares citados acima. Segundo os autores, *iLearning* fornece meios para auxiliar no aprendizado, por exemplo facilitando que usuários aprendam sozinhos ou em grupos de maneira mais fácil e a qualquer hora ou lugar. O *iSocial* tem foco na melhoria da comunicação informal dentro do campus; para os autores, esse pilar se divide em comunicação no currículo do campus, extracurriculares e atividades sociais. O pilar *iGreen* tem o carácter da sustentabilidade, em que a Internet das Coisas é usada para reduzir gastos e desperdícios, dentre outros serviços que tornem o campus

mais sustentável. *iHealth* é voltado a questão da saúde dos habitantes do campus, como auxiliar em alertas de epidemias ou mostrar informações do ambiente que possam causar algum dano a saúde dos habitantes. *iManagement* possui foco na segurança física do ambiente, como o monitoramento dos prédios ou de outros recursos do campus. Por último, temos o pilar *iGovernance*, o mesmo utiliza a IoT para auxiliar no gerenciamento da parte organizacional do campus, como por exemplo gestão de processos e gestão da administração.

Este trabalho terá como base o conceito de Campi Inteligentes, tendo em vista que o sistema é desenvolvido com o intuito de ser utilizado nas salas de aula da Universidade Federal do Ceará no campus de Quixadá. O sistema se alinha com o conceito de *iGreen* uma vez que irá controlar recursos como as lâmpadas das salas de aula diminuindo assim o tempo ocioso das mesmas e, conseqüentemente, reduzindo o desperdício de recursos públicos. Adicionalmente pelo fato do sistema desenvolvido neste trabalho propor o controle de presença dos alunos cadastrados no sistema, bem como o controle dos recursos através de informações cadastradas no sistema, este trabalho também se encaixa no contexto de *iGovernance*.

## **2.3 Tecnologias de Suporte**

Normalmente, o desenvolvimento ou criação de algo requer o uso de ferramentas e materiais. Com a Internet das Coisas não é diferente, para o desenvolvimento de aplicações na área, algumas tecnologias existentes são usadas como ferramenta, elas auxiliam em partes essenciais para as aplicações IoT, como: sensoriamento, comunicação e acesso as funcionalidades do sistema (KRUPITZER et al., 2015). As tecnologias de suporte estão inseridas dentro das camadas da arquitetura para aplicações IoT debatida na seção 3 deste trabalho. Logo a seguir serão apresentadas algumas dessas tecnologias importantes para o desenvolvimento de uma aplicação IoT, que são relevantes para o desenvolvimento deste trabalho. Dentre a variedade de tecnologias de suporte disponíveis, esta seção terá como foco a apresentação das tecnologias mais relevantes para este trabalho.

### **2.3.1 Hardware**

Nesta seção serão apresentados hardwares que servem como ferramenta para o desenvolvimentos de aplicações IoT. Dentre esses hardwares estão inseridos sensores, atuadores e placas de prototipagem.

**Sensores** são dispositivos com a capacidade de converter parâmetros físicos, químicos ou biológicos em sinais elétricos, de forma que dados digitais possam ser enviados para outros dispositivos com capacidade de processamento como uma placa de prototipagem Arduíno (NIST, 2009). Os sensores são usados em diferentes domínios como os definidos por Atzori, Iera e Morabito (2010), transporte e logística, saúde, ambiente inteligente, pessoal e social. Os **atuadores**, por sua vez, são dispositivos que recebem informações e as convertem para ações físicas (NIST, 2009). Em outras palavras podemos utilizar sensores para coletar informações e atuadores para, com base nessas informações, realizar ações para controlar um elemento da camada de ativos.

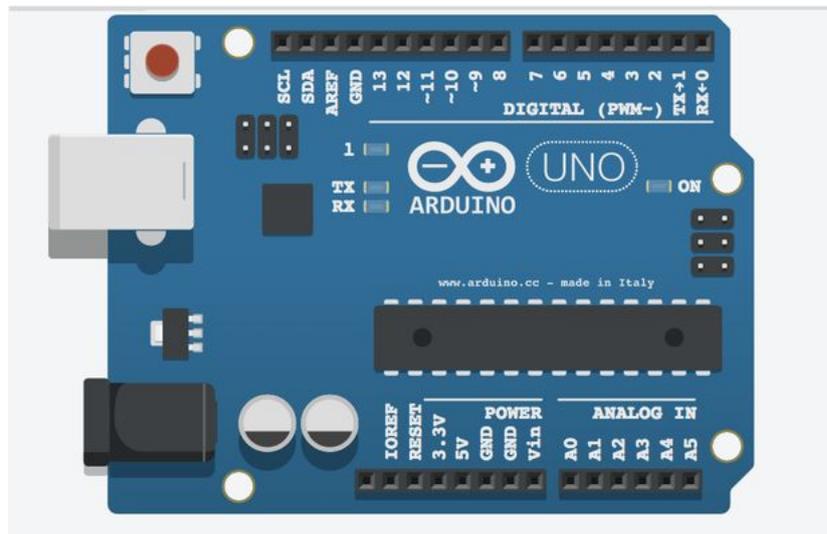
Um exemplo de utilização desses dispositivos é controle de um condicionador de ar, em que um sensor de temperatura receberia como parâmetro informações da temperatura do ambiente e enviaria essas informações para serem utilizadas com base para tomadas de decisões, um atuador receberia essas informações coletadas e seria capaz de controlar o condicionador de ar para manter a temperatura do ambiente agradável. Kortuem et al. (2010) afirmam que as redes de sensores, juntamente com atuadores e dispositivos com capacidade de processamento e armazenamento, evoluíram de dispositivos de detecção para objetos inteligentes que apoiam a criação de aplicações IoT.

As **placas de prototipagem** são placas pré-fabricadas que possuem microchips que facilitam a criação de protótipos para realização de testes sem a necessidade de construir placas de circuito impresso (JAMIESON; HERDTNER, 2015). Dentre as plataformas de prototipagem mais conhecidas, está o Arduíno, representado na Figura 2. Trata-se de uma plataforma de código aberto e de fácil uso do seu hardware e software. Por meio de um conjunto de instruções é possível informar a placa o que ela deve fazer (ARDUINO, 2017). Um exemplo de placa Arduíno é o UNO, que possui um microcontrolador ATmega328P com 16 MHz de velocidade de *clock* e 2KB de memória RAM. A placa ainda possui 14 pinos de entrada e saída, sendo 6 de saídas PWM e outras 6 de entradas analógicas. O Arduíno UNO é recomendado para quem deseja começar com eletrônica, pois é de fácil uso, basta conecta-lo a um computador com um cabo USB para começar a usar.(ARDUINO, 2017)

### **2.3.2 Comunicação**

Com base nas definições de Internet das Coisas apresentadas na seção 2.1 deste trabalho, percebe-se a importância da comunicação em aplicações IoT. Segundo Maia, Rocha

Figura 2 – Arduino UNO



Fonte – INSTRUCTABLES (2016).

e Andrade (2009), a comunicação é requisito básico para uma aplicação IoT. *Wi-Fi*, *LTE* e *Bluetooth* são exemplos de tecnologias para realizar a comunicação sem fio nessas aplicações. Normalmente, tecnologia de comunicação é escolhida no momento do desenvolvimento da aplicação, o que é um problema porque não existe uma tecnologia genérica indicada para todas as situações; em aplicações auto-adaptativas, é possível, se necessário, mudar a tecnologia para se adaptar a mudanças no ambiente de execução (MAIA et al., 2014). Nesta seção serão apresentadas algumas dessas tecnologias de comunicação sem fio que podem ser usadas em aplicações IoT.

*Radio Frequency Identification* ou apenas **RFID** é um sistema de identificação que usa *tags* ou etiquetas, em objetos a serem identificados. As etiquetas RFID são divididas em ativas e passivas, as ativas transmitem periodicamente sinais de ID, e as passivas são ativadas apenas quando estão na presença de algum leitor RFID (WEINSTEIN, 2005).

Uma outra tecnologia de comunicação bastante usada é a tecnologia *bluetooth*, um padrão para troca de dados em curta distância, criando redes de área pessoal confiáveis (BLUETOOTH, 2010). Desta tecnologia, surgiu o *Bluetooth Low Energy* ou **BLE**, essa tecnologia possui uma série de vantagens em relação ao *bluetooth* convencional, tornando assim o BLE uma das tecnologias mais recomendadas para aplicações IoT (BLUETOOTH, 2010). Dentre essas vantagens estão: baixo consumo de energia, envio de arquivos para diversos dispositivos ao mesmo tempo e capacidade de enviar pequenas quantidade de *bytes* em *broadcast* sem a necessidade de criar conexão manual com o dispositivo (BLUETOOTH, 2010).

Para este trabalho, são usados dispositivos *beacons*, que utilizam a tecnologia BLE para envio de informações para dispositivos em um raio de 100 metros (ZAFARI; PAPAPANAGIOTOU, 2015). Os *Beacons* funcionam como faróis que enviam mensagens com pequena quantidade de dados no formato de *broadcast* para dispositivos como *smartphones* e *tablets*, informando-os sobre a sua existência (STATLER, 2016). O sistema desenvolvido no presente trabalho usa o protocolo conhecido como *iBeacon*, desenvolvido pela empresa *Apple*. Este protocolo baseia-se na tecnologia BLE e tem como foco os dispositivos da própria empresa, onde os mesmos reagem as mensagens enviadas pelo *Beacon*. Uma das características determinantes para a escolha de utilizar *beacons* neste trabalho, foi a capacidade que os mesmos tem, de enviar os dados sem a necessidade de criar uma conexão com os dispositivos do usuário.

Na seção 4.1 é apresentado um trabalho que utiliza dispositivos *beacons* para realizar o controle de presença e compartilhamento de arquivos.

### 2.3.3 *Middleware*

*Middleware* é uma camada responsável por gerenciar o acesso aos recursos de hardware citados nesta seção, coletar dados produzidos pelos sensores e inferir informações de contexto (MAIA; ROCHA; ANDRADE, 2009). Esta camada utiliza as tecnologias de comunicação para realizar a troca de mensagens, abstraindo as camadas de baixo nível, possibilitando que os desenvolvedores de aplicações IoT possam se concentrar em funcionalidades de um nível mais alto. Entretanto, Autili et al. (2012) alertam que, devido ao fato de aplicações IoT serem normalmente imprevisíveis, a fronteira entre a aplicação e o *middleware* não é clara, sendo importante o uso de *middlewares* flexíveis que se adaptem ao comportamento da aplicação IoT.

O *middleware* é responsável por tornar sensores, atuadores e outros dispositivos endereçáveis e acessíveis através da internet. Quando esses dispositivos são acessados, eles precisam de soluções que possibilitem a sua comunicação, o *middleware* é uma dessas possíveis soluções. A camada de *middleware* também deve ser capaz de gerenciar os sensores e atuadores, fazendo com que os mesmos sejam descobertos em tempo de execução, dessa maneira a aplicação fica mais flexível e dados de fontes diferentes podem ser compostos em informações mais complexas. Além disso, dispositivos que por algum motivo ficaram indisponíveis, podem ser substituídos em tempo de execução. O *middleware* oferece interfaces entre aplicações, serviços e os hardwares da aplicação. Dessa maneira o sistema encapsula algumas informações de baixo

nível como citado anteriormente.

As capacidades da camada de *middleware* citadas anteriormente, tem grande importância neste trabalho, porque, com a abstração de recursos de baixo nível, o uso de um *middleware* facilita a comunicação entre os mesmos e as demais camadas do sistema.

### 3 OBJETIVOS

#### 3.1 Objetivo Geral

Este trabalho tem como **objetivo geral**, a construção de um sistema que auxilie na gerência de serviços das salas de aula da Universidade Federal do Ceará - Campus Quixadá, utilizando os conceitos de IoT e *Smart Campus*. O sistema será capaz de controlar alguns recursos das salas de aula como lâmpadas e gerenciar a presença dos alunos, de forma a auxiliar os professores, alunos e o Campus em geral.

#### 3.2 Objetivos específicos

Para atingir o objetivo geral deste trabalho de maneira satisfatória, foram definidos alguns objetivos específicos a serem atingidos, tais como:

- a) Desenvolver um sistema Web para gerenciar os recursos disponíveis e atualizar informações necessárias para o projeto.
- b) Criar um protótipo Arduino que controle os recursos das salas de aula através de um *Beacon* e relés.
- c) Implementar uma API, responsável por disponibilizar informações ao aplicativo.
- d) Desenvolver um aplicativo *mobile* que terá a função de atuador no sistema.
- e) Realizar testes do sistema em ambientes reais de execução.

## 4 TRABALHOS RELACIONADOS

### 4.1 GreatRoom: Uma Aplicação Android Baseada em Proximidade para a Criação de Salas Virtuais Inteligentes

Darin et al. (2016) utilizam os conceitos de Internet das Coisas e *Smart Offices* para desenvolver a aplicação móvel GreatRoom que, com base na proximidade dos usuários, cria salas virtuais e gerencia os serviços de presença e compartilhamento de arquivos. Darin et al. (2016) utilizam *beacons* para monitorar a entrada e saída de pessoas em um ambiente, e apontam algumas vantagens em usar esses dispositivos em seu trabalho, como: possuir uma maior precisão em ambientes internos, conseguir prover uma comunicação segura baseada em proximidade e a compatibilidade com uma grande quantidade de dispositivos. Os autores pretendem, com o uso do GreatRoom, eliminar a necessidade de autenticação de funcionários por meio de digital ou crachás, uma vez que seja possível verificar a movimentação de entrada e saída de cada pessoa em diferentes tipos de ambientes da empresa.

A aplicação usa os *smartphones* dos usuários como atuadores. Eles executam o aplicativo, que detecta e se conecta automaticamente ao *beacon* que está ao seu alcance através de uma conexão *bluetooth*, após realizada a conexão, são criadas salas virtuais com usuários próximos. Para que o usuário participe de uma sala virtual, o aplicativo troca mensagens com um serviço Web, que armazena essas informações na nuvem. Desta forma o usuário não precisa realizar qualquer requisição para ser inserido em uma sala, os dispositivos se comunicam automaticamente e realizam o *check-in* sem nenhuma interação direta com o usuário.

Darin et al. (2016) conseguem identificar as salas por meio de mensagens de *broadcast* enviadas pelos *beacons*, para realizar essa comunicação a aplicação utiliza o padrão de mensagens *Publisher-Subscriber*. O uso deste padrão, segundo os autores, minimiza o uso da largura da banda de rede, aumenta a confiabilidade da entrega das mensagens e é um princípio importante para aplicações de Internet das Coisas, devido os recursos limitados dos dispositivos.

A arquitetura da aplicação GreatRoom é bastante simples e possui três componentes principais: comunicação, serviço e apresentação. Os autores não entram em detalhes sobre os dispositivos usados, mas deixam claro quais tecnologias utilizaram e mostram como elas funcionam. Este trabalho se difere do GreatRoom pois o contexto do GreatRoom são os *Smart Offices* e a aplicação não utiliza *middleware*, apesar dos autores afirmarem que pretendem inseri-lo para realizar a comunicação dos sensores e atuadores com a aplicação. A aplicação

GreatRoom esta em fase de testes, Darin et al. (2016) propõem a resolução de uma problemática que também é abordada neste trabalho, o controle de presença dos usuários do sistema. Portanto as tecnologias e as soluções propostas por Darin et al. (2016) auxiliam na escolha das tecnologias utilizadas no desenvolvimento deste trabalho.

#### **4.2 Sistema inteligente de controle e monitoramento de ambiente de laboratórios de análises químicas**

Silva (2016) propõe a construção de um sistema para controlar, de forma inteligente, condicionadores de ar e desumidificadores em laboratórios de análises químicas. O protótipo do sistema utiliza os dados coletados pelos sensores distribuídos pelo ambiente, para identificar um padrão de comportamento. Silva (2016) utiliza algoritmos de aprendizado de máquina; dessa forma o autor agrupa os dados gerados pelos sensores e, com base em regras de controle, o sistema consegue gerenciar os condicionadores de ar e os desumidificadores. Desta forma, o protótipo mantém a temperatura e umidade do ambiente estáveis de forma inteligente.

O autor divide a construção da arquitetura do sistema em três etapas. Primeiramente foi construído um transmissor de dados, quando Silva (2016) utiliza a placa de prototipagem Arduíno UNO para coletar dados proveniente de cinco sensores de temperatura e cinco sensores de umidade espalhados pelo ambiente; além de coletar os dados, esta parte do projeto também é responsável por enviar essas informações para o banco de dados e por criar uma interface simples que mostra ao usuário informações sobre o ambiente através de leds e display. Em seguida, foi desenvolvido, um receptor de controle, composto de um Arduíno Mega que controla um Shield GSM/GPRS e um ESP8266; ele é responsável por receber dados do servidor e com base nessas informações, controlar o desumidificador e condicionador de ar via infra-vermelho. Na última etapa, de monitoramento remoto do ambiente, por meio de uma página web desenvolvida em PHP, são mostradas informações de temperatura e umidade do ambiente por meio de gráficos, com intuito de facilitar a visualização desses dados.

O autor utiliza uma ferramenta para criar um aplicativo simples que recebe informações do servidor, através de arquivos JSON e de maneira semelhante a página web, o aplicativo mostra dados sobre a temperatura e umidade do ambiente.

Apesar de Silva (2016) gerenciar condicionadores de ar e desumidificadores, diferentemente deste trabalho, ele detalha o uso dos dispositivos em sua arquitetura, bem como o funcionamento de cada um. Silva (2016) ainda mostra, que com o decorrer do seu projeto, ele

conseguiu gerar contribuições que auxiliaram no desenvolvimento do protótipo, como o aumento de portas na placa de prototipação Arduino. Portanto o trabalho de Silva (2016) traz informações importantes para este trabalho.

#### **4.3 Desenvolvimento de um sistema de automação residencial de baixo custo aliado ao conceito de Internet das Coisas (IoT)**

Com base no conceito de *Smart Homes*, Wanzeler, Fülber e Merlin (2016) desenvolvem um sistema de automação residencial de baixo custo controlado por meio de um dispositivo móvel. O sistema controla toda a iluminação de uma residência, de forma geral ou em ambientes específicos, o sistema também é capaz de monitorar a temperatura dos cômodos, se comunicando com o *smartphone* do usuário, afim de informa-lo que algo está ameaçando a segurança da sua residência, como um possível incêndio.

Os autores dividem o desenvolvimento do sistema em três partes principais: sistema de iluminação, monitoramento da temperatura e sistema de alarme residencial. No sistema de iluminação, o usuário pode desligar e ligar luzes, bem como controlar a intensidade da iluminação por meio de seu *smartphone*; Wanzeler, Fülber e Merlin (2016) afirmam em seu trabalho que essa comodidade pode gerar a redução do consumo de energia. Para a parte de monitoramento de temperatura, os autores utilizaram um sensor LM35; em seu protótipo os autores apenas monitoraram a temperatura de um determinado comodo, entretanto eles citaram possíveis melhorias como o controle da temperatura gerenciando dispositivos para resfriar ou aquecer o ambiente. Por fim, o sistema de alarme foi baseado na utilização de um sensor de movimento; após a ativação do alarme via aplicativo, o sistema monitora movimentos dentro da residência, caso algo movimente-se dentro do raio de captura do sensor, uma buzina emite um sinal sonoro e o usuário é notificado pelo *twitter*.

Ainda que os autores tenham como contexto principal as *Smart Homes* e gerencie diferentes recursos, Wanzeler, Fülber e Merlin (2016) propõem um sistema de baixo custo para gerenciar serviços de um determinado ambiente utilizando o conceito de Internet das Coisas.

Tabela 1 – Comparação entre os trabalhos relacionados

<b>Trabalhos</b>	<b>Foco</b>	<b>Domínio</b>	<b>Produto</b>
(DARIN et al., 2016)	Controle de presença e compartilhamento de arquivos.	<i>Smart Offices</i>	Protótipo
(SILVA, 2016)	Controle e monitoramento de ativos.	<i>Smart Environments</i>	Protótipo
(WANZELER; FÜLBER; MERLIN, 2016)	Automação residencial de baixo custo.	<i>Smart Homes</i>	Protótipo
Este trabalho	Controle de presença e ativos de salas de aula.	<i>Smart Campus</i>	Sistema

Fonte – Desenvolvida pelo Autor.

A tabela 1, faz a comparação entre os trabalhos apresentados nesta seção e o presente trabalho. A comparação leva em consideração o foco, domínio e o produto final de cada trabalho. Assim é possível visualizar de maneira mais rápida algumas diferenças entre os mesmos.

## 5 PROPOSTA

Este trabalho propõe o desenvolvimento de um sistema para gerenciar recursos das salas de aula da Universidade Federal do Ceará - Campus Quixadá. O sistema é baseado nos conceitos de IoT, com foco em *Smart Campus*, *iGreen* e *iGovernance*, apresentados na seção 2 de fundamentação teórica. O desenvolvimento da arquitetura, bem como os módulos do sistema tiveram como base informações retiradas tanto da seção 2 de fundamentação teórica, quanto da seção 4 de trabalhos relacionados.

### 5.1 Arquitetura do Sistema

Uma arquitetura, como explicado na seção 2 de fundamentação teórica, tem o intuito de mostrar os elementos concretos e conceituais que compõe o sistema. Na arquitetura proposta neste trabalho, os elementos concretos estão presentes na camada de ativos e recursos, a placa Arduíno é um exemplo. Além de mostrar os elementos presentes no sistema, a arquitetura também deixa claro como os mesmos se relacionam entre si.

Os principais módulos do sistema são o cliente, servidor, interceptador e placa. Os mesmos estão distribuídos pelas camadas de aplicação, comunicação, recursos e ativos, apresentados na Figura 3.

A seguir são especificados cada módulo, bem como as tecnologias utilizadas no desenvolvimento dos mesmos.

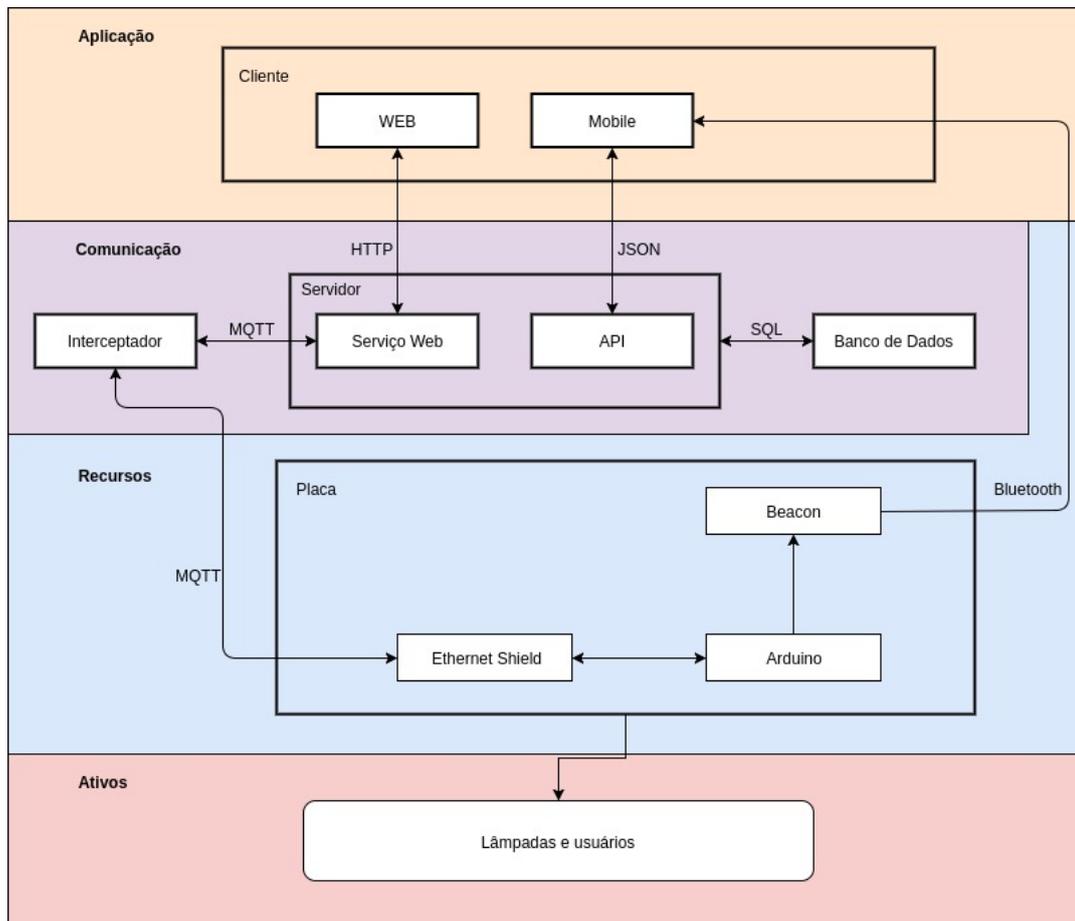
#### 5.1.1 Cliente

O módulo cliente é dividido em web e *mobile*, através desse módulo os usuários podem interagir com o sistema, seja adicionando, atualizando ou consumindo informações. Apesar de ambos serem a porta de comunicação do usuário com o sistema, eles possuem diferenças que serão especificadas a seguir.

##### 5.1.1.1 Web

Através do cliente web, usuários com permissões específicas podem gerenciar grande parte das informações presentes no sistema. Por exemplo, é através deste módulo que um usuário pode ligar e desligar as luzes de uma determinada sala de aula, ou ativar e desativar um *Beacon*. O

Figura 3 – Arquitetura do sistema



Fonte – Desenvolvida pelo Autor.

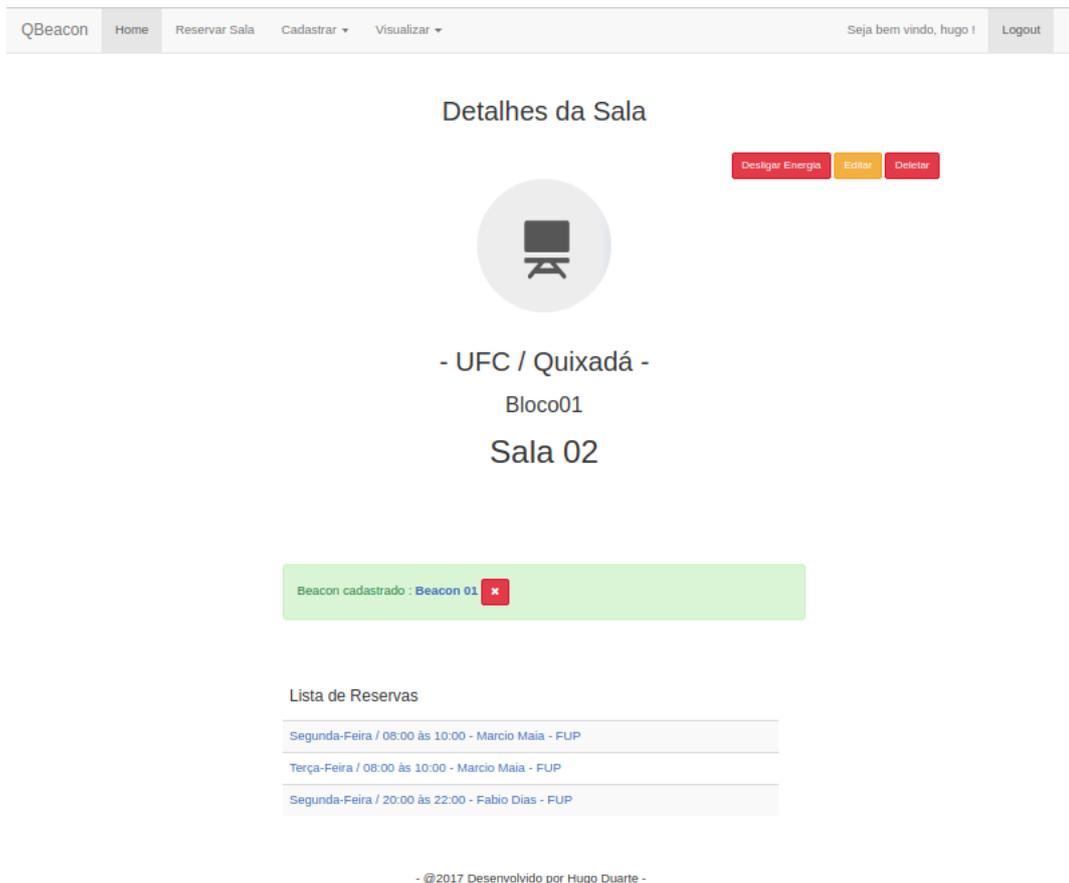
cliente web se comunica diretamente com o serviço web, recebendo e enviando dados utilizando o protocolo HTTP (*Hypertext Transfer Protocol*).

O desenvolvimento do cliente foi realizado utilizando as ferramentas: HTML5, JavaScript, Thymeleaf e CSS. Para agilizar o processo de estilização dos componentes foi utilizado o *framework Bootstrap*<sup>3</sup>.

Para exemplificar algumas funcionalidades do cliente web, a Figura 4 mostra a tela com informações específicas de uma determinada sala de aula, como: nome, bloco, campus e instituição a qual pertence. Nesta tela também é apresentado ao usuário a lista de reservas da sala, informando o dia da semana, horário e a turma de cada reserva. A tela ainda dá ao usuário as opções de editar ou deletar a sala, desligar e ligar as luzes da mesma e remover ou adicionar um *Beacon* previamente cadastrado no sistema.

<sup>3</sup> <<https://getbootstrap.com/>>

Figura 4 – Tela de detalhes da sala



Fonte – Desenvolvida pelo Autor.

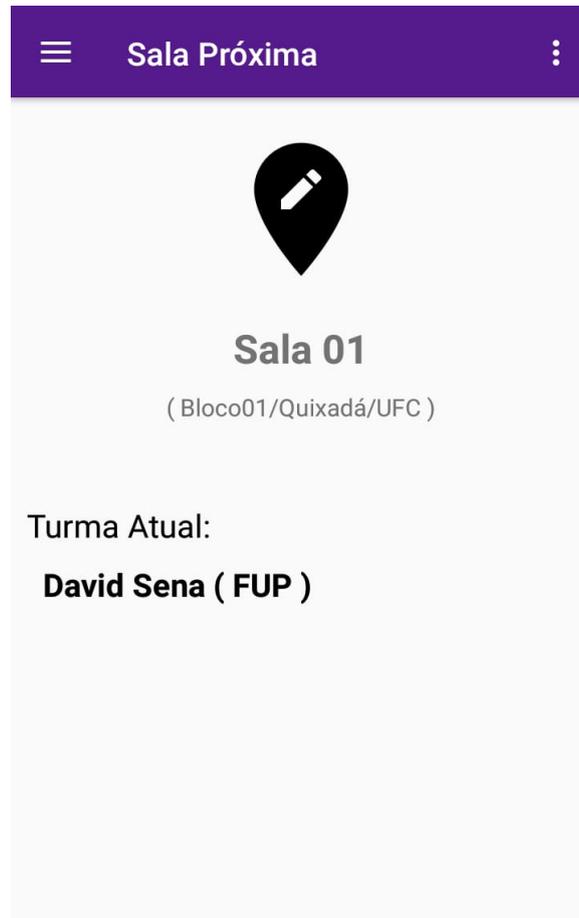
### 5.1.1.2 Mobile

A aplicação *mobile* é desenvolvida para *smartphones* com o sistema operacional Android e os seus principais usuários são os alunos. A aplicação é responsável por apresentar aos usuários informações importantes do sistema. Ao utilizar a aplicação *mobile* o usuário pode visualizar todas as disciplinas, salas e turmas cadastradas no sistema, bem como informações detalhadas de cada uma. A aplicação consome essas informações, em formato de JSON (*JavaScript Object Notation*) que são disponibilizadas pela API (*Application Programming Interface*), a subseção 5.1.2.2 contém maiores detalhes sobre a API desenvolvida neste trabalho. Além de ter acesso a informações, através do aplicativo o usuário pode participar de uma determinada turma, caso isso aconteça o usuário passa a ser aluno daquela turma e a mesma é adicionada a lista de turmas daquele usuário.

Uma das funcionalidades mais importantes da aplicação *mobile* é apresentada na Figura 5. A figura mostra a tela onde o usuário consegue visualizar informações de uma sala a

qual o seu dispositivo esteja próximo. Para isso a aplicação *mobile* usa a biblioteca *AltBeacon*<sup>4</sup>, esta biblioteca permite que dispositivos Androids consigam trabalhar com *iBeacons* de maneira semelhante a dispositivos *IOS*.

Figura 5 – Tela de sala próxima



Fonte – Desenvolvida pelo Autor.

Após encontrar um *Beacon* utilizando a biblioteca *AltBeacon*, apresentada anteriormente, a aplicação extrai o UUID (*Universally Unique Identifier*) do *Beacon* encontrado. É importante salientar que o UUID de um *Beacon* é utilizado para identificação do mesmo, porém neste trabalho, a primeira faixa de UUID é utilizada para identificar a sala a qual o *Beacon* pertence, sendo que o dígito mais significativo dessa faixa representa o status atual do *Beacon*, essas informações são detalhadas na seção 5.1.4. Tendo acesso ao UUID, a aplicação extrai o ID da sala, com isso, é realizada uma requisição a API, que por sua vez, retorna um JSON com informações da sala e da turma atual. O Código Fonte 1 é um trecho do código em Java referente ao processo descrito.

<sup>4</sup> <<https://altbeacon.github.io/android-beacon-library/>>

### Código-fonte 1 – Obtendo ID da sala através da mensagem do *Beacon*

```

1 public void onBeaconFinded(final Beacon b) {
2     runOnUiThread(new Runnable() {
3         String UUID = b.getId1().toUuid().toString().substring(0, 8);
4         if(UUID.charAt(0) != '2'){
5             String salaID = "";
6             @Override
7             public void run() {
8                 for (int i=1; i<UUID.length(); i++){
9                     char c = UUID.charAt(i);
10                    if(c != '0'){
11                        salaID = UUID.substring(i, UUID.length());
12                        break;
13                    }
14                }
15            }
16        }
17    });
18 }

```

Fonte – Desenvolvida pelo Autor.

A aplicação *mobile* ainda verifica, através do status do *Beacon*, se a turma atual na sala está realizando a presença de seus alunos. Em caso afirmativo, as informações do usuário são enviadas para a API e a mesma fica responsável por registrar a presença do usuário.

#### 5.1.2 Servidor

Este módulo é responsável por tratar as requisições do módulo cliente, fazer a comunicação do cliente com o interceptador e persistir os dados produzidos pelo sistema. Ele é dividido em serviço Web e API, ambos utilizam o servidor Apache Tomcat<sup>5</sup> e foram desenvolvidos usando o *Framework Spring Boot*<sup>6</sup>.

##### 5.1.2.1 Serviço Web

O serviço web, além de tratar as requisições HTTP (*Hypertext Transfer Protocol*), persistir e disponibilizar informações ao cliente web, ainda é responsável pela comunicação com o interceptador, controlando assim os elementos da camada de ativos.

<sup>5</sup> <<http://tomcat.apache.org/>>

<sup>6</sup> <<https://spring.io/projects/spring-boot>>

Para a comunicação com o interceptador o serviço web utiliza o protocolo MQTT (*Message Queue Telemetry Transport*)<sup>7</sup>, bastante utilizado na comunicação M2M (*machine-to-machine*). Este protocolo implementa a arquitetura *publish/subscriber*, onde os clientes podem enviar e receber informações através de tópicos (LAMPKIN et al., 2012).

O serviço web usa como cliente MQTT o *Paho Java Client*<sup>8</sup> e o Código Fonte 2 mostra um método simples que publica uma mensagem em um determinado tópico utilizando este cliente.

#### Código-fonte 2 – Publicando uma mensagem usando Paho Java Client

```

1 public void publicar(String mensagem, String topico) throws MqttException {
2     MqttClient client = new MqttClient(Constants.URI_MQTT, MqttClient.
        generateClientId());
3     client.connect();
4     MqttMessage message = new MqttMessage();
5     message.setPayload(mensagem.getBytes());
6     client.publish(topico, message);
7 }

```

Fonte – Desenvolvida pelo Autor.

A gerencia dos elementos da camada de ativos é realizada pelo serviço web de duas formas, através de ações diretas do usuário por meio da aplicação web ou de forma automatizada com base nas informações cadastradas no sistema. De forma direta o usuário pode, por exemplo, utilizar a aplicação web para enviar uma requisição solicitando que as lâmpadas de uma determinada sala sejam desligadas. O serviço web por sua vez recebe esta requisição e publica no tópico referente ao controle desses recursos uma mensagem para realizar a ação solicitada. O Código Fonte 3 mostra a implementação da funcionalidade citada.

#### Código-fonte 3 – Requisição para desligar as luzes de uma sala

```

1 @GetMapping("/desligar/{id_sala}")
2 public String desligarEnergia(@PathVariable("id_sala") Integer id_sala) {
3     Sala sala = salaService.buscarSala(id_sala);
4     if(sala.isEnergia()) {
5         String mqttTopico = (sala.getBloco().getCampus().getInstituicao().getNome() +
            "/"
6             + sala.getBloco().getCampus().getNome() + "/"
7             + sala.getBloco().getNome() + "/"

```

<sup>7</sup> <<http://mqtt.org/>>

<sup>8</sup> <<https://www.eclipse.org/paho/clients/java/>>

```

8         + sala.getNome().toUpperCase();
9     try {
10        mqttTopico = Normalizer.normalize(mqttTopico, Normalizer.Form.NFD).
            replaceAll("[^\\p{ASCII}]", "")
11        .replaceAll(" ", "");
12        this.publicar("0", mqttTopico);
13        sala.setEnergia(false);
14        salaService.salvarSala(sala);
15    } catch (Exception e) {
16        // TODO: handle exception
17    }
18
19 }
20 return "redirect:/sala/" + sala.getId();
21 }

```

Fonte – Desenvolvida pelo Autor.

Para gerenciar de forma automática os elementos da camada de ativos, o sistema utiliza informações cadastradas na aplicação web e persistidas no banco de dados. As ações para a gerencia desses elementos são executadas de forma assíncrona e tomam como base, além das informações cadastradas no sistema, o dia da semana, horário atual e os horários das aulas da Universidade Federal do Ceará - Campus Quixadá. As mesmas ocorrem nos horários fixos de 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 de segunda à sexta. Com essas informações o sistema está apto a executar de forma automática as ações de ligar e desligar as lâmpadas das salas. Essa ação ocorre de forma semelhante ao apresentado no Código Fonte 3. A outra ação é desligar e ligar o modo de presença do *Beacon*, isso significa alterar o dígito mais significativo do UUID para que ao receber as informações do *Beacon*, a aplicação *mobile* saiba que pode enviar os dados do usuário para que seja realizado a presença do mesmo em uma determinada aula. A Tabela 2 mostra as ações a serem realizadas e as condições para que elas sejam executadas.

Tabela 2 – Ações executadas de forma automática pelo sistema

Ações	Condições para execução
Ligar lâmpadas	Sala possuir aula. 15 minutos antes de iniciar a aula.
Desligar lâmpadas	Sala não possuir aula. 15 minutos depois de terminar a aula.
Ligar modo presença do <i>Beacon</i>	Sala possuir aula. 15 minutos depois de iniciar a aula.
Desligar modo presença do <i>Beacon</i>	Sala possuir aula. 25 minutos depois de iniciar a aula.

Fonte – Desenvolvida pelo Autor.

### 5.1.2.2 API

A API ou *Application Programming Interface* é responsável por persistir e disponibilizar informações à aplicação mobile, bem como tratar as requisições enviadas pela mesma. O desenvolvimento da API neste trabalho possibilita uma maior flexibilidade principalmente no envio de dados no formato JSON. Dessa forma, a API disponibiliza apenas as informações realmente necessárias, diminuindo o fluxo de dados no sistema. A API foi na arquitetura REST (*Representational State Transfer*), que por sua vez utiliza o HTTP como protocolo de comunicação através dos métodos *GET*, *POST*, *PUT* e *DELETE*.

### 5.1.3 Interceptador

O interceptador utilizado no sistema é o *broker* Eclipse Mosquitto<sup>9</sup>, ele é responsável por enviar as informações do serviço web para a placa Arduino na camada de recursos. A comunicação do interceptador, tanto com o serviço web, quanto com a camada de recursos é realizada utilizando o protocolo MQTT e a arquitetura *publish/subscriber*. Quando o serviço web publica uma mensagem em um determinado tópico o interceptador é responsável por interceptar a mensagem e enviar para a placa Arduino que tiver subscrito naquele determinado tópico. Especificamente neste sistema, cada Arduino se inscreve em dois tópicos distintos, um tópico é responsável por controlar o *Beacon* e o outro para controlar os recursos da sala de aula. Os tópicos nos quais o Arduino se inscreve seguem o padrão mostrado no Código Fonte 4, onde o tópico responsável por controlar o *Beacon* é o nome do *Beacon* em letras maiúsculas, sem espaços e sem caracteres especiais. O tópico responsável por controlar os recursos da sala de aula

<sup>9</sup> <<https://mosquitto.org/>>

é a junção dos nomes da **Instituição**, **Campus**, **Bloco** e **Sala**, essas informações são escritas em letras maiúsculas, sem espaços e sem caracteres especiais, separados por '/

#### Código-fonte 4 – Arduíno se inscrevendo em tópicos

```

1 #define BEACONTOPIC "BEACON01"
2 #define SALATOPIC "UFC/QUIXADA/BLOCO01/SALA01"
3
4 void reconnect() {
5     while (!client.connected()) {
6         Serial.print("Tentando uma conexão MQTT...");
7         if (client.connect("arduinoClient")) {
8             Serial.println("Conectado!");
9             client.subscribe(BEACONTOPIC);
10            client.subscribe(SALATOPIC);
11        }
12        else {
13            Serial.print("Erro ao se conectar, rc=");
14            Serial.print(client.state());
15            Serial.println("Tentando conectar novamente em 5s ...");
16            delay(5000);
17        }
18    }
19 }

```

Fonte – Desenvolvida pelo Autor.

O Código Fonte 4 mostra a definição dos tópicos para controlar o *Beacon* e para controlar as lâmpadas, a definição segue o padrão apresentado anteriormente. No mesmo código fonte, a função *reconnect* é responsável por conectar o Arduíno ao interceptador e se inscrever nos tópicos definidos.

#### 5.1.4 Placa

Este módulo é desenvolvido utilizando uma placa de prototipação Arduíno, a mesma é responsável por alterar os estados das lâmpadas e do *Beacon*. O Arduíno se conecta ao interceptador e inscreve em dois tópicos, como apresentado no Código Fonte 4, ao receber uma determinada mensagem, o Arduíno difere de qual tópico veio a mensagem e com base na informação recebida, executa uma determinada ação, o Código Fonte 5 mostra como é implementada esta funcionalidade.

#### Código-fonte 5 – Função executada ao receber uma mensagem do interceptador

```

1 void callback(char* topic , byte* payload , unsigned int length) {
2   String mensagem = "";
3   String topico = "";
4
5   for (int i=0;i<(int)strlen(topic);i++) {
6     topico += topic[i];
7   }
8   for (int i=0;i<length;i++) {
9     mensagem += (char)payload[i];
10  }
11  if(topico == BEACONTOPIC){
12    ble.setUUID(0 , mensagem.substring(0 , 8));
13  }
14  if(topico == SALATOPIC){
15    if(mensagem == "1"){
16      digitalWrite(led , HIGH);
17      delay(3000);
18    }
19    if(mensagem == "0"){
20      digitalWrite(led , LOW);
21      delay(3000);
22    }
23  }
24 }

```

Fonte – Desenvolvida pelo Autor.

Ao receber uma mensagem no tópico responsável pela comunicação com o *Beacon*, o Arduíno altera a primeira faixa do UUID do mesmo, com isso as informações enviadas pelo *Beacon* são atualizadas. O UUID de um *Beacon* é composto por 4 faixas de 8 dígitos hexadecimais, neste trabalho a primeira faixa é utilizada para o envio de informações do status do *Beacon* e da sala de aula ao qual o mesmo está conectado. O dígito mais significativo representa o status do *Beacon* e o restante dos dígitos são utilizados para enviar o ID da sala de aula. A tabela 3 mostra os dígitos referentes a cada status possíveis dos *Beacons* neste trabalho.

Tabela 3 – Possíveis status de um *Beacon*

<b>Dígito mais significativo</b>	<b>Status</b>
0	Ativo Enviando informações de uma sala de aula
1	Ativo Enviando informações de uma sala de aula Realizando a presença de usuários próximos
2	Desativado Não esta enviando informações de nenhuma sala

Fonte – Desenvolvida pelo Autor.

Caso o Arduino receba uma mensagem com o conteúdo igual a **0** no tópico responsável pela comunicação com os recursos da sala de aula, o Arduino desligará as lâmpadas da sala de aula, caso a mensagem tenha o conteúdo igual a **1**, as lâmpadas serão ligadas pelo Arduino. Para realizar essas ações a placa utiliza um dispositivo conhecido como relé, responsável por interromper a corrente elétrica a qual o mesmo esteja inserido, dessa forma o Arduino consegue ligar ou desligar as lâmpadas de uma sala de aula que estão ligadas à corrente controlada pelo relé. O Código Fonte 5 mostra o Arduino alterando o status do relé com base nas informações recebidas pela função de *callback*.

## 6 RESULTADOS E DISCUSSÕES

Esta seção apresenta os resultados obtidos no trabalho de implementação de um sistema para gerência de serviços das salas de aula da Universidade Federal do Ceará - Campus Quixadá, bem como as dificuldades encontrados durante o desenvolvimento do mesmo.

Desenvolver um sistema que tem como base os conceitos de Internet das Coisas é um grande desafio, apesar de ser uma tecnologia bastante estudada, utilizar placas de prototipação e diferentes dispositivos requer bastante cuidado e pesquisa. Algumas configurações mudam conforme o fabricante e podem ocasionar problemas no desenvolvimento do protótipo. Isso ocorreu neste projeto ao utilizar um módulo *Bluetooth* com as especificações recomendadas para o uso como *iBeacon*, porém o mesmo não respondia a alguns comandos necessários para que fosse utilizado no projeto. A Figura 6 coloca lado a lado os dois módulos *Bluetooth* utilizados neste trabalho, o módulo da esquerda não respondeu aos comandos como esperado, já o módulo da direita, do fabricante SunFounder<sup>10</sup>, respondeu os comandos e foi utilizado no projeto. A tabela 4 mostra as especificações de ambos módulos.

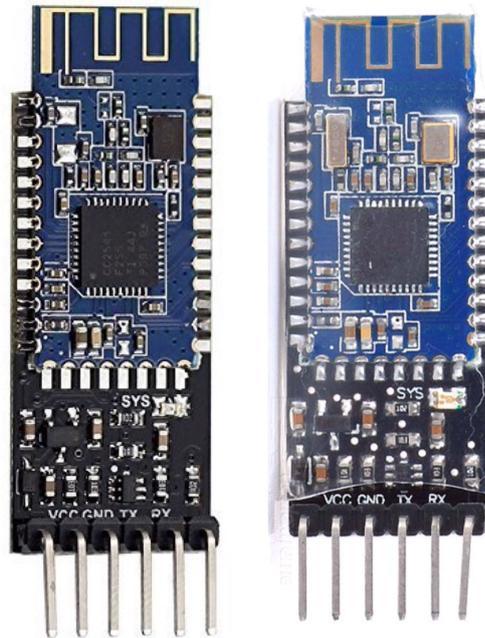
Tabela 4 – Especificações dos módulos *Bluetooth*

Especificações	
Chipset	CC2541
Versão	V4.0 Ble
Banda	2.4GHz ISM
Tensão de Operação	3.3/5V
Corrente de Funcionamento	50mA
Alcance	100 metros em campo aberto

Fonte – Desenvolvida pelo Autor.

Dificuldades já eram esperadas no decorrer do desenvolvimento do projeto, entretanto foi possível cumprir os objetivos específicos apresentados na seção 3.2. Neste projeto foi desenvolvido um sistema web, composto por com uma aplicação web que se comunica diretamente com um serviço web e possibilita o usuário gerenciar recursos das salas de aula; Um protótipo Arduíno que recebe informações do serviço web através do interceptador e com base nessas informações altera o estado das lâmpadas e do *Beacon* de uma sala; Uma API que recebe requisições da aplicação *mobile* e disponibiliza as informações persistidas no banco de dados do sistema; Uma aplicação *mobile* que é responsável por apresentar as informações

<sup>10</sup> <<https://www.sunfounder.com/bluetooth-4-0-hm-10-master-slave-module.html>>

Figura 6 – Módulos *Bluetooth*

Fonte – Desenvolvida pelo Autor.

disponibilizadas pela API aos usuários, a mesma ainda é responsável por enviar informações do usuário para a gerência da presença dos mesmos.

O sistema conseguiu atingir o objetivo de gerenciar as lâmpadas das salas de aula, a tela da aplicação web que permite esse controle é apresentada na Figura 4 e uma parte da implementação desta funcionalidade é disponibilizada no Código Fonte 3. Além do controle manual das lâmpadas, com base em informações cadastradas na aplicação web, o sistema consegue gerenciar de forma automática esses recursos. A Figura 7 mostra a tela onde o usuário pode reservar uma determinada sala, essa informação é de extrema importância pois com ela é possível saber se uma sala possui ou não aula em um determinado horário.

O objetivo de gerenciar a presença dos alunos utilizando *Beacons* também foi atingido neste trabalho. Com base na tabela 2, caso o *Beacon* esteja com o modo presença ativado, os usuários que estejam próximos a esse *Beacon*, através da aplicação *mobile*, enviam uma requisição com os dados do usuário para a API, que por sua vez fica responsável por realizar a presença do usuário. Para melhorar a controle sobre esse serviço, utilizando a aplicação web, é possível editar a lista de alunos presentes em uma determinada aula, para isso o usuário deve ir nos detalhes da aula. O Código Fonte 6 mostra a implementação da funcionalidade que adiciona um usuário a lista de alunos presentes em uma determinada aula. A Figura 5 apresenta a tela da aplicação *mobile* responsável por mostrar ao usuário informações da sala de aula próxima ao seu

Figura 7 – Tela de reserva de uma sala

Fonte – Desenvolvida pelo Autor.

dispositivo.

### Código-fonte 6 – Método para realizar presença de um usuário

```

1 @PostMapping(path="{ id_turma }/presenca/{ id_reserva}")
2 public MensagemRetorno darPresenca(@RequestBody @Valid Usuario usuario ,
3   @PathVariable("id_turma") Integer id_turma , @PathVariable("id_reserva")
4   Integer id_reserva) throws Exception{
5   try {
6     Turma turmaBanco = turmaService.buscarTurma(id_turma);
7     Reserva reservaBanco = reservaService.buscarReserva(id_reserva);
8     String data= this.getData();
9     Aula aulaBanco = aulaService.buscarAula(turmaBanco.getId(), data);
10    if(aulaBanco == null) {
11      aulaBanco = this.criaAula(turmaBanco, data, reservaBanco);
12    }
13    this.adicionaAlunoFrequencia(usuario, aulaBanco);
14
15    return new MensagemRetorno(Constants.SUCESSO_PRESENCA_USUARIO);
16  } catch (Exception e) {
17    throw new Exception(Constants.ERRO_PRESENCA_USUARIO);
18  }

```

Fonte – Desenvolvida pelo Autor.

## 6.1 Avaliação

Na fase de avaliação do sistema foram encontradas algumas dificuldades, como a falta de tempo e recursos para realizar os testes com mais placas de prototipação Arduino. Não poder realizar a avaliação do sistema nos ambientes reais das salas de aula da Universidade Federal do Ceará - Campus Quixadá também foi uma dificuldade enfrentada nesta fase do trabalho.

A avaliação da funcionalidade de gerência das lâmpadas e *Beacons* das salas de aula, não puderam ser realizada nas salas de aula da Universidade Federal do Ceará - Campus Quixadá. Entretanto, os testes das lâmpadas foram realizados utilizando um circuito semelhantes ao utilizado nas salas de aula do campus. Os testes foram realizados de maneira manual, utilizando a aplicação web e através da gerência automática desses recursos realizada pelo serviço web, com base nas informações cadastradas no sistema. No primeiro caso as ações possíveis seriam de desligar e ligar as lâmpadas de uma sala de aula, bem como ativar e desativar um determinado *Beacon*. Em relação a estes testes o sistema se comportou como esperado, enviando mensagens via MQTT para o Arduino, que por sua vez, com base nas informações recebidas realizava as ações de alterar os estados do relé, ligando e desligando as lâmpadas e modificando o UUID do *Beacon*. Nos testes relacionados a gerência automática das lâmpadas e *Beacons*, as ações possíveis eram as de desligar e ligar as lâmpadas de uma sala de aula, bem como ativar e desativar o modo presença de um *Beacon*. O sistema também se comportou de maneira satisfatória, para agilizar o processo de testes dessa funcionalidade foram passados diferentes dias e horários, com base nessas informações o sistema conseguiu realizar as ações programadas nas regras de negócio.

A funcionalidade de gerenciar a presença dos usuários utilizando o *Beacon* foi avaliada em três dispositivos Android diferentes, **Moto g2**, **Xiaomi Redmi 4X** e **LG X power**, a Tabela 5 mostra as especificações de cada dispositivo. Todos os dispositivos tiveram as permissões de uso dos recursos de *Bluetooth* e *Fine Location* ativadas. Apenas o dispositivo **Xiaomi Redmi 4X** não conseguiu encontrar o *Beacon*, não foi possível descobrir o motivo de tal problema, mas o sistema operacional deste dispositivo é modificado e pode ocasionar alguma

incompatibilidade com a biblioteca *AltBeacon*, utilizada para localizar os *Beacons*. Os demais dispositivos utilizados, **Moto g2** e **LG X power**, conseguiram encontrar o *Beacon* e visualizar as informações enviadas no UUID do mesmo. Ao encontrar o *Beacon*, os dispositivos foram capazes de visualizar as informações da sala de aula referente ao *Beacon* e enviar uma requisição para que a API realizasse a presença do usuário.

Tabela 5 – Especificações dos dispositivos

<b>Dispositivo</b>	<b>SO</b>	<b>Bluetooth</b>	<b>GPS</b>
Xiaomi Redmi 4X	Android 6.0.1 MIUI v8	4.2 com A2DP/LE	A-GPS/GLONASS/BeiDou
LG X power	Android 6.0.1	4.2 com A2DP/LE	A-GPS/GLONASS
Moto G2	Android 6.0	4.0 com A2DP/LE	A-GPS/GLONASS

Fonte – Desenvolvida pelo Autor.

## 7 CONSIDERAÇÕES FINAIS

Com o desenvolvimento deste projeto é possível observar como a Internet das Coisas pode ser utilizada para potencializar os benefícios e as funcionalidades de um sistema. O projeto desenvolvido poderia ser utilizado apenas como um sistema comum para gerenciar as salas de aulas, turmas, disciplinas e outras informações da Universidade Federal do Ceará - Campus Quixadá, ou qualquer outra instituição que tenha uma estrutura semelhante. Entretanto, com o uso da IoT, é possível controlar e otimizar o uso de recursos físicos do campus, auxiliando assim professores, alunos e o campus em geral.

Atingindo os objetivos específicos de desenvolver um sistema web para gerenciar o uso das lâmpadas e *Beacons* das salas de aula, criar um protótipo Arduino para realizar o controle desses recursos, implementar uma API e desenvolver uma aplicação *mobile*, foi possível alcançar o objetivo geral almejado, desenvolvendo um sistema para gerenciar os serviços das salas de aula da Universidade Federal do Ceará - Campus Quixadá.

Como melhorias, seguem algumas sugestões para trabalhos futuros:

- Adicionar as funcionalidades de controlar os condicionadores de ar e os projetores, assim o sistema ficaria responsável por gerenciar grande parte dos recursos das salas de aula.
- Verificar a distância do *Beacon* para os dispositivos e utilizar esses dados para verificar se o usuário está dentro da sala de aula.
- Evoluir o sistema para um *framework*, deixando-o mais genérico e facilitando o seu reuso, bem como a implementação de novas funcionalidades.

Os códigos apresentados neste trabalho, estão disponíveis em:  
<https://github.com/fhugoduarte>

## REFERÊNCIAS

- AARTS, E.; WICHERT, R. Ambient intelligence. In: **Technology Guide**. [S.l.]: Springer, 2009. p. 244–249.
- ARDUINO. **What is Arduino?** 2017. Acessado em: 19 maio 2017. Disponível em: <<https://www.arduino.cc/>>.
- ASHTON, K. That ‘internet of things’ thing. **RFiD Journal**, v. 22, n. 7, p. 97–114, 2009.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.
- AUTILI, M.; INVERARDI, P.; PELLICCIONE, P.; TIVOLI, M. Developing highly complex distributed systems: a software engineering perspective. **Journal of Internet Services and Applications**, Springer, v. 3, n. 1, p. 15–22, 2012.
- BLUETOOTH, S. Bluetooth core specification version 4.0. **Specification of the Bluetooth System**, 2010.
- COOK, D.; DAS, S. K. **Smart environments: Technology, protocols and applications**. [S.l.]: John Wiley & Sons, 2004. v. 43.
- DARIN, T.; BARBOSA, J.; RODRIGUES, B.; ANDRADE, R. Greatroom: Uma aplicação android baseada em proximidade para a criação de salas virtuais inteligentes. In: **Workshop de Ferramentas e Aplicações (WFA). Simpósio Brasileiro de Sistemas Multimídia e Web**. [S.l.: s.n.], 2016.
- EVANS, D. The internet of things how the next evolution of the internet is changing everything (april 2011). **White Paper by Cisco Internet Business Solutions Group (IBSG)**, 2012.
- 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. **BeginnerArduino**. 2016. Acessado em: 19 maio 2017. Disponível em: <<http://www.instructables.com/id/Beginner-Arduino/>>.
- JAMIESON, P.; HERDTNER, J. More missing the boat—arduino, raspberry pi, and small prototyping boards and engineering education needs them. In: IEEE. **Frontiers in Education Conference (FIE), 2015. 32614 2015. IEEE**. [S.l.], 2015. p. 1–6.
- KORTUEM, G.; KAWSAR, F.; SUNDRAMOORTHY, V.; FITTON, D. Smart objects as building blocks for the internet of things. **IEEE Internet Computing**, IEEE, v. 14, n. 1, p. 44–51, 2010.
- KRUPITZER, C.; ROTH, F. M.; VANSYCKEL, S.; SCHIELE, G.; BECKER, C. A survey on engineering approaches for self-adaptive systems. **Pervasive and Mobile Computing**, Elsevier, v. 17, p. 184–206, 2015.

LAMPKIN, V.; LEONG, W. T.; OLIVERA, L.; RAWAT, S.; SUBRAHMANYAM, N.; XIANG, R.; KALLAS, G.; KRISHNA, N.; FASSMANN, S.; KEEN, M. et al. **Building smarter planet solutions with mqtt and ibm websphere mq telemetry**. [S.l.]: IBM Redbooks, 2012.

LANDT, J. The history of rfid. **IEEE potentials**, IEEE, v. 24, n. 4, p. 8–11, 2005.

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.

MAIA, M. E.; ANDRADE, R. M.; FILHO, C. A. de Q.; BRAGA, R. B.; AGUIAR, S.; MATEUS, B. G.; NOGUEIRA, R.; TOORN, F. Usable—a communication framework for ubiquitous systems. In: IEEE. **Advanced Information Networking and Applications (AINA), 2014 IEEE 28th International Conference on**. [S.l.], 2014. p. 81–88.

MAIA, M. E.; ROCHA, L. S.; ANDRADE, R. Requirements and challenges for building service-oriented pervasive middleware. In: ACM. **Proceedings of the 2009 international conference on Pervasive services**. [S.l.], 2009. p. 93–102.

MINERVA, R.; BIRU, A.; ROTONDI, D. Towards a definition of the internet of things (iot). **IEEE Internet Initiative**, n. 1, 2015.

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.

NIE, X. Constructing smart campus based on the cloud computing platform and the internet of things. In: **the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013)**. [S.l.: s.n.], 2013.

NIST. **What is a sensor?** 2009. Acessado em: 18 maio 2017. Disponível em: <<https://www.nist.gov/el/intelligent-systems-division-73500/definitions>>.

SANTOS, G. A. D.; MACHADO, G. J. C.; JUNIOR, R. A. D. A.; SOUZA, M. S. D. Internet of things (iot): Um cenário guiado por patentes industriais. **GESTÃO. Org: Revista Eletrônica de Gestão Organizacional**, v. 13, 2015.

SILVA, A. R. d. **Sistema inteligente de controle e monitoramento de ambiente de laboratórios de análises químicas**. Dissertação (Mestrado) — Instituto Metrópole Digital, Universidade Federal do Rio Grande do Norte, Natal, 2016.

STATLER, S. Geofencing: Everything you need to know. In: **Beacon Technologies**. [S.l.]: Springer, 2016. p. 307–316.

WANZELER, T.; FÜLBER, H.; MERLIN, B. Desenvolvimento de um sistema de automação residencial de baixo custo aliado ao conceito de internet das coisas (iot). **XXXIV Simpósio Brasileiro de Telecomunicações. Santarém, PA**, p. 40–44, 2016.

WEINSTEIN, R. Rfid: a technical overview and its application to the enterprise. **IT professional**, IEEE, v. 7, n. 3, p. 27–33, 2005.

ZAFARI, F.; PAPAPANAGIOTOU, I. Enhancing ibeacon based micro-location with particle filtering. In: IEEE. **Global Communications Conference (GLOBECOM), 2015 IEEE**. [S.l.], 2015. p. 1–7.

ZANELLA, A.; BUI, N.; CASTELLANI, A.; VANGELISTA, L.; ZORZI, M. Internet of things for smart cities. **IEEE Internet of Things journal**, IEEE, v. 1, n. 1, p. 22–32, 2014.