



**UNIVERSIDADE ESTADUAL DO MARANHÃO**  
**CENTRO DE CIÊNCIAS TECNOLÓGICAS**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DA COMPUTAÇÃO E**  
**SISTEMAS**  
**MESTRADO PROFISSIONAL EM ENGENHARIA DA COMPUTAÇÃO E SISTEMAS**  
**THIAGO NUNES DE SOUSA**

**ARQUITETURA E CLIENTE PARA GRAVAÇÃO DE ÁUDIOS DE AULAS POR**  
**MEIO DE DISPOSITIVOS DE INTERNET DAS COISAS**

**SÃO LUÍS – MA**

**2017**

THIAGO NUNES DE SOUSA

ARQUITETURA E CLIENTE PARA GRAVAÇÃO DE ÁUDIOS DE AULAS POR MEIO DE  
DISPOSITIVOS DE INTERNET DAS COISAS

Dissertação apresentada ao Curso de Mestrado Profissional em Engenharia da Computação e Sistemas do Programa de Pós-Graduação em Engenharia da Computação e Sistemas do Centro de Ciências Tecnológicas da Universidade Estadual do Maranhão, como requisito parcial à obtenção do título de mestre em Engenharia da Computação e Sistemas. Área de Concentração: Tecnologia da Informação

Orientador: Reinaldo de Jesus da Silva

Co-Orientador: Luís Carlos Fonseca

SÃO LUÍS – MA

2017

Sousa, Thiago Nunes de.

Arquitetura e cliente para gravação de áudios de aulas por meio de dispositivos de internet das coisas. / Thiago Nunes de Sousa. – São Luís, 2017.

68 f.

Dissertação (Mestrado) – Programa de Pós-graduação em Engenharia da Computação e Sistemas, Universidade Estadual do Maranhão, 2017.

Orientador: Prof. Dr. Reinaldo de Jesus da Silva.

1. Internet das coisas. 2. Arquitetura orientada a serviços. 3. Informática na educação. 4. Raspberry Pi. I. Título.

CDU 004:37

THIAGO NUNES DE SOUSA

ARQUITETURA E CLIENTE PARA GRAVAÇÃO DE ÁUDIOS DE AULAS POR MEIO DE  
DISPOSITIVOS DE INTERNET DAS COISAS

Dissertação apresentada ao Curso de Mestrado Profissional em Engenharia da Computação e Sistemas do Programa de Pós-Graduação em Engenharia da Computação e Sistemas do Centro de Ciências Tecnológicas da Universidade Estadual do Maranhão, como requisito parcial à obtenção do título de mestre em Engenharia da Computação e Sistemas. Área de Concentração: Tecnologia da Informação

Aprovada em: 15 de Agosto de 2017

BANCA EXAMINADORA

---

Reinaldo de Jesus da Silva (Orientador)  
Universidade Estadual do Maranhão – UEMA

---

Luís Carlos Fonseca (Co-Orientador)  
Universidade Estadual do Maranhão – UEMA

---

Fernando Jorge Cutrim Demétrio  
Universidade Estadual do Maranhão – UEMA

---

Sofiane Labidi  
Universidade Federal do Maranhão – UFMA

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

“É melhor lançar-se à luta em busca do triunfo mesmo expondo-se ao insucesso, que formar fila com os pobres de espírito, que nem gozam muito nem sofrem muito; E vivem nessa penumbra cinzenta sem conhecer nem vitória nem derrota.”

(Franklin Roosevelt)

## **AGRADECIMENTOS**

Primeiramente a Deus que permitiu que tudo isso acontecesse, ao longo de minha vida, e não somente nestes anos como estudante, mas que em todos os momentos é o maior mestre que alguém pode conhecer.

Agradeço a todos os professores, em especial Reinaldo de Jesus da Silva e Luís Carlos Fonseca, orientador e co-orientador respectivamente, por me proporcionarem o conhecimento não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, por tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

À esta universidade, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbro um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.

Aos meus pais e irmã, pelo amor, incentivo e apoio incondicional.

À minha namorada, que nos momentos de minha ausência dedicados ao estudo, sempre fez entender que o futuro é feito a partir da constante dedicação no presente!

## RESUMO

Propõe-se neste trabalho, um sistema computacional que utilize dispositivos de internet das coisas como ferramenta de suporte a professores e alunos, realizando a gravação de áudios de aulas, o registro de anotações e a disponibilização online automática destes recursos. É utilizada uma arquitetura orientada a serviços, através de uma API REST, que garante a interoperabilidade entre dispositivos heterogêneos e fornece os serviços necessários para o cliente Web e para o cliente de gravação, Raspberry Pi, além de arquitetura de banco de dados redundante com MongoDB. Espera-se que, com o sistema proposto, o processo de revisão e elucidação de dúvidas dos alunos seja facilitado, através da reprodução dos áudios gravados e da consulta às anotações da turma, em momento posterior à aula. Tal expectativa é confirmada através da aplicação de questionários a alunos do curso de Engenharia da Computação, da Universidade Estadual do Maranhão, após contato com o protótipo desenvolvido, e os resultados obtidos são trazidos neste trabalho.

**Palavras-chave:** Internet das Coisas. Arquitetura Orientada a Serviços. Informática na Educação. Raspberry Pi.



## **ABSTRACT**

It is proposed in this work, a computer system that uses internet of things devices as a tool to support teachers and students, performing the recording of classroom audios, recording of notes and automatic upload of these resources. A service-oriented architecture is used through a REST API that ensures interoperability between heterogeneous devices and provides the necessary services for the Web client and the recording client, Raspberry Pi, as well as a redundant database architecture with MongoDB. It is expected, with the proposed system, the process of review and elucidation of students doubts will be facilitated, through the reproduction of the recorded audios and the consultation to the annotations of the class, in a moment after the lesson. This expectation is validated by the application of questionnaires to students of the course of Computing Engineering of Universidade Estadual do Maranhão, after contact with the developed prototype, and the results obtained are brought in this work.

**Keywords:** Internet of Things. Service Oriented Architecture. Computer Education. Raspberry Pi.

## LISTA DE FIGURAS

Figura 1 – Internet das Coisas como o resultado da convergência de diferentes visões . . . . .	17
Figura 2 – Hiper-Ciclo de Tecnologias Emergentes . . . . .	18
Figura 3 – Ciclo de Vida de um Serviço Web SOAP . . . . .	21
Figura 4 – Arquitetura Proposta . . . . .	31
Figura 5 – Diagrama de Casos de Uso Básicos . . . . .	32
Figura 6 – Exemplo de Chamada e Retorno da API . . . . .	34
Figura 7 – Diagrama de Sequência do mecanismo de autenticação . . . . .	35
Figura 8 – Topologia do ambiente de Banco de Dados . . . . .	36
Figura 9 – Diagrama de Casos de Uso . . . . .	39
Figura 10 – Interface para Iniciar e Encerrar Sessão . . . . .	40
Figura 11 – Interface de detalhamento da sessão . . . . .	41
Figura 12 – Diagrama de Estado do Cliente Web . . . . .	42
Figura 13 – Diagrama de Estado do Dispositivo de Gravação . . . . .	42
Figura 14 – Diagrama de Sequência da Gravação de Áudio . . . . .	44
Figura 15 – Interface Web em Smartphone . . . . .	45
Figura 16 – Dispositivo e seus periféricos . . . . .	46
Figura 17 – Respostas da primeira pergunta . . . . .	48
Figura 18 – Respostas da segunda pergunta . . . . .	49
Figura 19 – Respostas da terceira pergunta . . . . .	49
Figura 20 – Respostas da quarta pergunta . . . . .	50
Figura 21 – Respostas da quinta pergunta . . . . .	50
Figura 22 – Respostas da sexta pergunta . . . . .	51
Figura 23 – Respostas da sétima pergunta . . . . .	51

## LISTA DE TABELAS

Tabela 1 – Comparação entre o E-things e os trabalhos relacionados . . . . .	26
Tabela 2 – Resultado dos questionários aplicados . . . . .	52

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	13
1.1	OBJETIVOS	14
<b>1.1.1</b>	<b>Objetivo Geral</b>	14
<b>1.1.2</b>	<b>Objetivos Específicos</b>	15
1.2	APRESENTAÇÃO DO TRABALHO	15
<b>2</b>	<b>REFERENCIAL TEÓRICO</b>	16
2.1	COMPUTAÇÃO UBÍQUA	16
2.2	EDUCAÇÃO UBÍQUA	16
2.3	INTERNET DAS COISAS	17
<b>2.3.1</b>	<b>Web das Coisas</b>	19
2.4	ARQUITETURA ORIENTADA A SERVIÇOS	19
<b>2.4.1</b>	<b>SOAP</b>	20
<b>2.4.2</b>	<b>REST</b>	20
2.5	CONSIDERAÇÕES DO CAPÍTULO	22
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	24
3.1	CONSIDERAÇÕES DO CAPÍTULO	26
<b>4</b>	<b>METODOLOGIA</b>	28
4.1	SUJEITOS DA PESQUISA	28
4.2	INSTRUMENTO DE COLETA DE DADOS	29
4.3	PROCEDIMENTOS PARA COLETA DE DADOS	29
4.4	RECURSOS TECNOLÓGICOS	30
4.5	CONSIDERAÇÕES DO CAPÍTULO	30
<b>5</b>	<b>PROTOTIPAGEM DO SISTEMA PROPOSTO</b>	31
5.1	ARQUITETURA DA PLATAFORMA PROPOSTA	31
<b>5.1.1</b>	<b>Requisitos Básicos do Sistema</b>	32
<b>5.1.2</b>	<b>Arquitetura do Software</b>	33
<b>5.1.3</b>	<b>Ambiente de Banco de Dados</b>	34
<b>5.1.4</b>	<b>Infraestrutura de Servidores</b>	37
<b>5.1.5</b>	<b>Considerações sobre a plataforma</b>	37
5.2	CLIENTE DE GRAVAÇÃO DE AULAS	38
<b>5.2.1</b>	<b>Casos de Uso</b>	38

5.2.1.1	Iniciar Sessão . . . . .	38
5.2.1.2	Encerrar Sessão . . . . .	39
5.2.1.3	Upload da Gravação . . . . .	40
5.2.1.4	Adicionar, Visualizar Anotação e Reproduzir Áudio . . . . .	40
<b>5.2.2</b>	<b>Estados . . . . .</b>	<b>40</b>
<b>5.2.3</b>	<b>Serviços Web Desenvolvidos . . . . .</b>	<b>43</b>
<b>5.2.4</b>	<b>Sincronismo das Mensagens . . . . .</b>	<b>44</b>
<b>5.2.5</b>	<b>Clientes Desenvolvidos . . . . .</b>	<b>45</b>
5.2.5.1	Cliente Web . . . . .	46
5.2.5.2	Cliente de Gravação . . . . .	46
<b>5.2.6</b>	<b>Considerações finais sobre o Cliente de Gravação . . . . .</b>	<b>47</b>
<b>6</b>	<b>RESULTADOS . . . . .</b>	<b>48</b>
6.1	RESULTADOS DO QUESTIONÁRIO . . . . .	48
6.2	CONSIDERAÇÕES SOBRE OS RESULTADOS . . . . .	52
<b>7</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS . . . . .</b>	<b>53</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>55</b>
	<b>APÊNDICES . . . . .</b>	<b>57</b>
	APÊNDICE A – Formulário Aplicado aos Alunos . . . . .	58
	APÊNDICE B – Definição da API . . . . .	61
	APÊNDICE C – Código-Fonte dos métodos para iniciar gravação e realizar envio da mídia no Raspberry Pi . . . . .	64
	APÊNDICE D – Código-Fonte dos <i>end-points</i> da API para envio e recupera- ção de gravações . . . . .	66

## 1 INTRODUÇÃO

A Internet das Coisa (*Internet of Things*), ou IoT, é um novo paradigma onde estamos cercados por uma variedade de objetos, ou coisas, capazes de interagir e cooperar entre si, a fim de atingir um objetivo comum (GIUSTO *et al.*, 2010). Sistemas de informação e comunicação compostos por estes objetos são fortemente incorporados ao ambiente em que vivemos, de forma que, em certas circunstâncias, se tornam até invisíveis (GUBBI *et al.*, 2013).

Neste cenário, vislumbra-se uma série de aplicações possíveis para diferentes áreas, como a automação, telemedicina, logística, transportes e educação, por exemplo. Na mesma velocidade, é preciso lidar com desafios novos inseridos pela heterogeneidade dos objetos, é necessário trabalhar com ambientes interoperáveis, seguros e confiáveis, tendo hardwares com escassez de recursos computacionais.

Por sua vez, a computação ubíqua, ou pervasiva, se dá pela criação de um ambiente naturalmente tomado pela capacidade de processamento e comunicação, de forma integrada ao usuário e sempre presente. Sistemas desta natureza tem características próprias que devem ser observadas, como o uso efetivo de espaços inteligentes, tecnologias integradas ao ambiente de maneira minimamente invasiva, escalabilidade para suportar a sobrecarga de uso e a capacidade de adaptação a ambientes diferentes (SATYANARAYANAN, 2001).

A rápida evolução das tecnologias de informação e comunicação trouxeram uma série de mudanças aos processos de aprendizagem. Hoje a educação não é mais restrita exclusivamente ao ambiente da escola, as pessoas tem acesso a todo tipo de conteúdo para estudo a qualquer momento e em todo lugar, inclusive através de dispositivos de Internet das Coisas (XUE; WANG; CHEN, 2011).

Imaginamos cenários onde um sensor possa captar o áudio ou as imagens de uma aula e automaticamente disponibilizá-los no AVA, onde estas mesmas aulas possam ser revisadas e comentadas por comando de voz através da central multimídia de um carro, onde materiais de apoio são criados e consumidos a qualquer momento através de dispositivos móveis, relógios ou óculos, onde residências possam se integrar a ambientes de aprendizado através de seus sistemas de automação, onde o conhecimento chegue ao aluno através de buscas inteligentes ou seja criado por ele, como postagens em redes sociais, um ambiente onde indivíduos, softwares e hardwares cooperam para a disseminação do conhecimento.

Alguns AVA's disponíveis no mercado, apesar de não terem sido concebidos com este fim, oferecem a possibilidade de conexão via serviços web, permitindo assim a comunicação

com dispositivos da Internet das Coisas através desta tecnologia. No entanto, uso da IoT trás consigo uma série de outros desafios, tais quais a interoperabilidade, segurança, confiabilidade, hardwares com poucos recursos computacionais, massiva geração de dados e pervasividade, necessitando assim de um ambiente pensado especificamente para atender à estas necessidades.

Estas possibilidades motivam o desenvolvimento de ambientes educacionais específicos para a Internet das Coisas que interajam com tecnologia educacionais, tais quais fóruns, chats, blogs, microblog, linha do tempo, redes sociais, dentre vários outros. Vislumbra-se que o caminho para obtenção deste nível de interoperabilidade, considerando redes sociais, dispositivos de internet das coisas e AVA, se torna viável através da utilização de uma arquitetura interoperável, escalável e com características pervasivas, permitindo a comunicação de qualquer tipo de objeto através de uma camada de serviços web.

Metodologicamente esta pesquisa caracteriza-se como bibliográfica e experimental e, quanto a sua natureza, como uma pesquisa aplicada. Frente à abordagem do problema, tem-se uma pesquisa qualitativa, já quanto aos seus objetivos esta é descritiva e explicativa. Em relação ao método de abordagem do problema, utiliza-se o método indutivo, de forma que, a partir de dados constatados inicialmente, infere-se generalizações e relações entre eles.

Esta pesquisa traz discussões sobre a utilização da internet das coisas integrada à informática na educação. É estudada a utilização de sistemas computacionais que se comportem de maneira pervasiva aplicados à educação, proporcionando um ecossistema onde entes reais e virtuais cooperem para a disseminação do conhecimento. Propõe-se aqui, uma plataforma educacional para a gravação de áudios de aulas, pensada especificamente para internet das coisas, que contempla um modelo de postagens genéricas e uma arquitetura que atende aos requisitos inerentes a este tipo de ambiente.

## 1.1 OBJETIVOS

Os objetivos deste encontram-se divididos entre objetivo geral e objetivos específicos.

### 1.1.1 Objetivo Geral

O objetivo geral desta pesquisa é propor um sistema computacional que utilize dispositivos de Internet das Coisas como ferramenta de suporte a professores e alunos com a gravação de áudios de aulas.

### 1.1.2 Objetivos Específicos

Para se atingir ao objetivo geral, esta pesquisa trabalha com os seguintes objetivos específicos:

Realizar levantamento bibliográfico das principais características e requisitos de sistemas de Internet das Coisas.

Identificar e analisar os fundamentos da área de computação ubíqua, em especial a sua aplicação em ambientes educacionais.

Apresentar a arquitetura da plataforma proposta, que dê suporte ao desenvolvimento de aplicações de Internet das Coisas para a educação.

Levantar os requisitos e propor sistema para gravação e disponibilização online de áudios de aulas, bem como registros de anotações, utilizando dispositivo de Internet das Coisas.

Apresentar protótipo e especificações do sistema para gravação de aulas e registro de anotações.

Disponibilizar a professores e alunos, em ambiente controlado, acesso ao protótipo desenvolvido do sistema.

Verificar, do ponto de vista técnico, se os requisitos elencados são atendidos pelo protótipo durante a utilização dos usuários.

Analisar a satisfação dos usuários em relação ao sistema proposto do ponto de vista dos áudios gravados e anotações realizadas.

## 1.2 APRESENTAÇÃO DO TRABALHO

Este trabalho está dividido em sete capítulos, os quais abordam os seguintes temas: o primeiro capítulo apresenta os conceitos iniciais de Internet das Coisas, computação e aprendizagem ubíqua, bem como os cenários para aplicações destas novas tecnologias; o objetivo geral, que delimita o escopo do estudo e facilita a sua orientação; os objetivos específicos, que especificam as etapas necessárias para o atingimento do objetivo geral. O segundo capítulo trás o referencial teórico e expõem de maneira mais detalhada alguns alguns conceitos já tratados, bem com apresenta a arquitetura orientada a serviço e suas tecnologias. No terceiro capítulo tem-se os trabalhos relacionados à esta pesquisa. O quarto capítulo apresenta a metodologia adotada para o desenvolvimento do trabalho. No quinto capítulo tem-se a prototipagem do sistema proposto, enquanto o sexto capítulo exhibe a pesquisa realizada, os resultados obtidos e as discussões. No último capítulo, o sétimo, são apresentadas as conclusões e trabalhos futuros.



## 2 REFERENCIAL TEÓRICO

### 2.1 COMPUTAÇÃO UBÍQUA

A computação ubíqua, ou pervasiva, acontece por meio da criação de um ambiente naturalmente tomado pela capacidade de processamento e comunicação, de forma integrada ao usuário e sempre presente (SATYANARAYANAN, 2001). Para Araujo (2003) “a ideia básica da computação ubíqua é que a computação move-e para fora das estações de trabalho e computadores pessoais (PCs) e torna-se pervasiva em nossa vida cotidiana”.

A computação ubíqua proporciona ambientes computacionais onde as pessoas interajam continuamente com centenas de “computadores” interconectados. Tal integração entre dispositivos e usuários chega ao ponto da tecnologia se tornar invisível, não necessariamente no sentido físico, mas de maneira que haja um relacionamento, um convívio, natural entre usuário e computador (WEISER, 1993).

O desenvolvimento de sistemas desta natureza apresenta problemas próprios que devem ser observados. É preciso transformar espaços físicos em espaços inteligentes, dotados de capacidade computacional para tomada de decisão e adaptáveis a situações imprevistas. Tecnologias presentes nestes ambientes devem se apresentar de maneira minimamente invasiva, de forma que o usuários as utilize inconscientemente. Outro ponto a se preocupar é a escalabilidade, é necessário que se desenvolvam arquiteturas que deem suporte à grande quantidade de processamento decorrente do crescimento das interações entre usuários e dispositivos (SATYANARAYANAN, 2001).

### 2.2 EDUCAÇÃO UBÍQUA

A rápida evolução das tecnologias de informação e comunicação trouxeram uma série de mudanças aos processos de aprendizagem. Hoje a educação não é mais restrita exclusivamente ao ambiente da escola, as pessoas tem acesso a todo tipo de conteúdo para estudo a qualquer momento e em todo lugar, inclusive através de dispositivos de Internet das Coisas (XUE; WANG; CHEN, 2011).

Para Santaella (2013), aprendizagem ubíqua são “as novas formas de aprendizagem mediadas pelos dispositivos móveis”, sendo que estes dispositivos móveis são “qualquer equipamento ou periférico que pode ser transportado com informação que fique acessível em qualquer lugar”. Para a autora, a evolução dos dispositivos móveis chegou ao ponto de “tornar

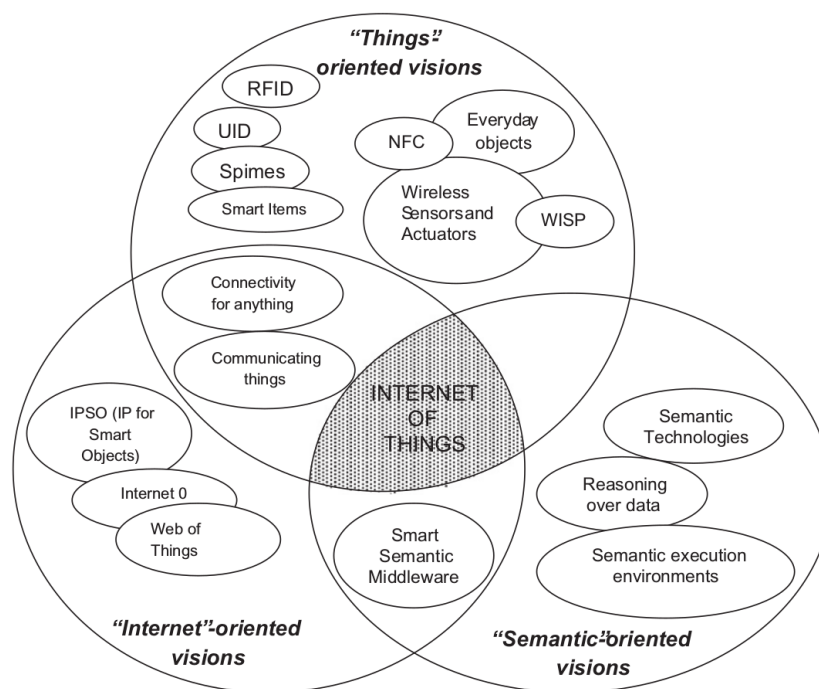
absolutamente ubíquos e pervasivos o acesso à informação, à comunicação e à aquisição de conhecimento”.

### 2.3 INTERNET DAS COISAS

A Internet das Coisas é um novo paradigma onde estamos cercados por uma variedade de objetos, ou coisas, capazes de interagir e cooperar entre si, a fim de atingir um objetivo comum. Dentre estas coisas tem-se etiquetas RFID (*Radio-Frequency IDentification*), sensores, atuadores, celulares, entre outros, que utilizam um mecanismo de endereçamento único para se comunicarem (GIUSTO *et al.*, 2010).

Os sistemas de informação e comunicação compostos por estes objetos são fortemente incorporados ao ambiente em que vivemos, de forma que, em certas circunstâncias, se tornam até invisíveis. Com isto, tem-se a geração de uma enorme quantidade de dados que precisam ser armazenados, processados e apresentados ao usuário de maneira adequada. Uma solução possível para estes problemas é através da utilização de serviços de computação na nuvem, através de ferramentas de armazenamento, análise e visualização de dados (GUBBI *et al.*, 2013).

Figura 1 – Internet das Coisas como o resultado da convergência de diferentes visões

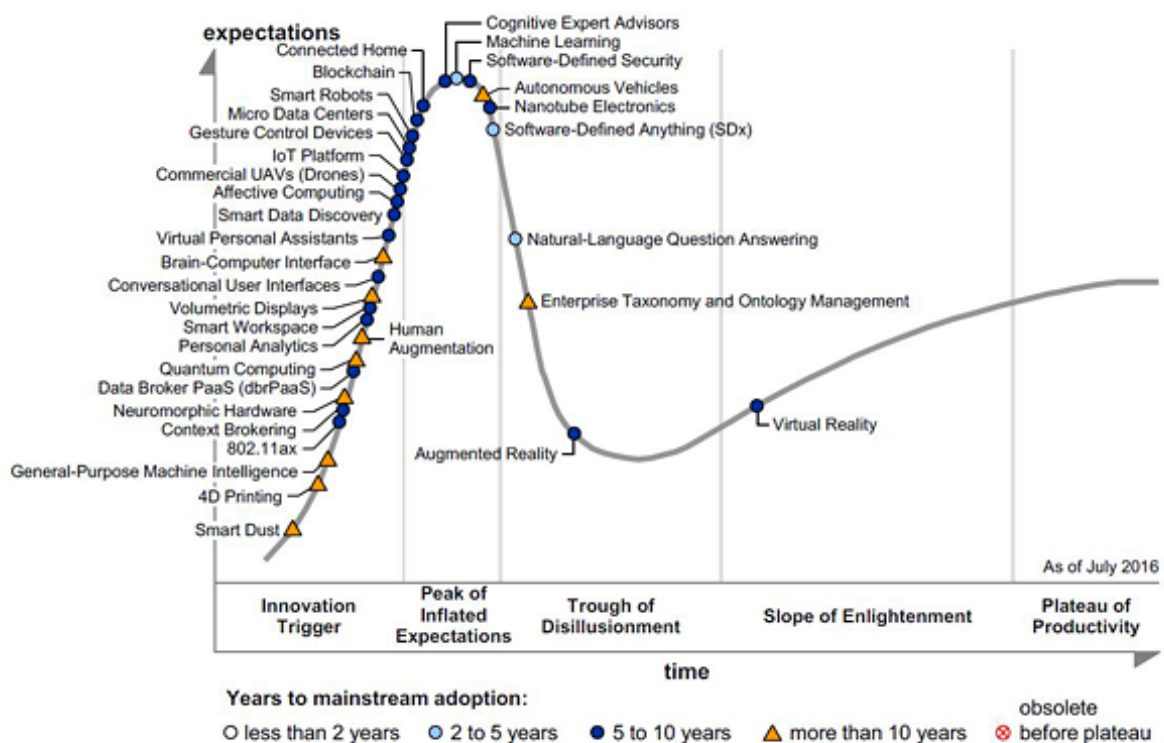


Fonte: (ATZORI; IERA; MORABITO, 2010)

Quanto à composição de sistemas para a Internet das Coisas, observa-se a distinção do problema em três visões, a da internet, a das coisas e a da semântica. O relacionamento entre estas três visões pode ser observado na Figura 1. A primeira diz respeito aos dispositivos (sensores, atuadores e etc) e os desafios atrelados a eles. A segunda, visa dirimir os problemas relacionados à comunicação, endereçamento e conectividade. Por último, a visão semântica traz a modelagem de soluções semânticas para a representação, armazenamento, busca e organização das informações geradas (ATZORI; IERA; MORABITO, 2010).

Para Miorandi *et al.* (2012), na próxima década, a Internet das Coisas propiciará que conteúdo e serviços estejam sempre ao nosso redor, sempre disponíveis, permitindo novas formas de trabalho, de interação, de entretenimento e de estilo de vida. Isto confirma-se em estudo do Gartner Inc. (2016), onde é previsto que a Internet das Coisas estará massificada no cotidiano das pessoas em cerca de 5 a 10 anos. Os dados desta pesquisa estão apresentados na Figura 2.

Figura 2 – Hiper-Ciclo de Tecnologias Emergentes



Fonte: (GARTNER INC., 2016)

À medida que este cenário se consolida, temos lidar com novos desafios decorrentes da heterogeneidade dos objetos, é preciso trabalhar com ambientes interoperáveis, seguros e

confiáveis tendo hardwares com escassez de recursos computacionais. Dentre as propostas para o enfrentamento destas dificuldades temos a utilização de uma Arquitetura Orientada a Serviços, uma camada de software que esconde dos objetos detalhes da aplicação. Softwares com estas características foram propostos por (GUINARD *et al.*, 2010), (KOVATSCHEK; MAYER; OSTERMAIER, 2012), (SPIESS *et al.*, 2009), (HEO *et al.*, 2015), (PINTUS; CARBONI; PIRAS, 2012), entre outros.

### 2.3.1 Web das Coisas

Objetos da internet das coisas (sensores, atuadores, RFID, etc) em grande parte não utilizam tecnologias compatíveis entre si, quando produzidos por fabricantes diferentes. Mesmo aqueles conectados à internet, geralmente precisam de um software específico para seu controle e gerenciamento. Este comportamento dificulta o desenvolvimento de aplicações compostas por diferentes objetos, que cruzem dados de diferentes sensores e possam atuar de maneira realmente integrada (GUINARD *et al.*, 2010).

Como solução para este problema de integração, Guinard e Trifa (2009) propõe a utilização de protocolos Web como padrão para a comunicação. Tal solução faz com que todos os dispositivos utilizem a mesma linguagem para troca de mensagens, bem como torna muito mais fácil a integração entre objetos do mundo real com recursos da Web.

## 2.4 ARQUITETURA ORIENTADA A SERVIÇOS

Segundo (SOMMERVILLE, 2011) “um web service é uma instância de uma ideia mais geral de um serviço”, onde um serviço é um meio de entregar resultado ao solicitante sem que este seja responsável pela execução. Eles foram propostos para permitirem a integração entre sistemas diferentes, da mesma organização ou não, por isso, têm como forte característica, a independência de plataforma. Segundo (ZIMMERMANN; TOMLINSON; PEUSER, 2012) são um meio simples, de baixo custo e multiplataforma para compartilhamento de funções e dados entre aplicações corporativas.

A utilização destes serviços para o desenvolvimento de sistemas deu surgimento à uma nova disciplina da engenharia de software, a Arquitetura Orientada a Serviços. Nela os sistemas são projetados para utilizarem recursos disponibilizados por web services, como por exemplo, um serviço de consulta à previsão do tempo ou de acesso aos contracheques da empresa. Esta abordagem traz vantagens no que tange à modularidade e reusabilidade, visto que

cada serviço tem seu escopo bem delimitado e as responsabilidades são encapsuladas e divididas.

Com o objetivo de prover todas estas características, a W3C definiu um padrão baseado em XML <sup>1</sup> para o desenvolvimento de web services, composto basicamente por um protocolo para encapsulamento de mensagens (SOAP), um para a descrição do serviço (WSDL) e outro para descoberta (UDDI). Em contraproposta a este padrão, surgiram os serviços baseados em REST, que utilizam exclusivamente o protocolo HTTP para a troca de mensagens e fornecimento de recursos e, segundo (LIU *et al.*, 2015), trouxeram inúmeras facilidades para nossa vida e trabalho. A baixo trataremos sobre estas duas tecnologias, suas características, vantagens e desvantagens.

### 2.4.1 SOAP

SOAP, *Simple Object Access Protocol*, é a tecnologia baseada em XML especificada pelo *World Wide Web Consortium* para o desenvolvimento de *web services*. Segundo o (W3C, 2000), é um protocolo leve destinado à troca de informações estruturadas em um ambiente descentralizado. Foi projetado para ser um protocolo de comunicação da camada de aplicação, logo, pode utilizar HTTP ou SMTP para transportar suas mensagens através da Internet.

Todas as mensagens SOAP devem ser obrigatoriamente encapsuladas por um envelope, que deve conter um cabeçalho (header) e um corpo (body), sendo que este último possui de fato a informação a ser transmitida.

Em um projeto de web service SOAP, é comum a utilização de outros dois protocolos complementares, o WSDL (*Web Services Description Language*) e o UDDI (*Universal Description, Discovery and Integration*), ambos baseados em XML. O primeiro especifica a interface do serviço, métodos, tipo de dados, mensagens trocadas, entre outros, enquanto o segundo funciona como um catálogo de serviços e fornece um meio para a descoberta de novos *web services*. Na Figura 3 é possível observar como se dão as interações entres estes protocolos, o cliente busca um serviço acessando a um UDDI, em seguida baixa o WSDL do provedor escolhido e por último acesas ao serviço web enviando mensagens SOAP.

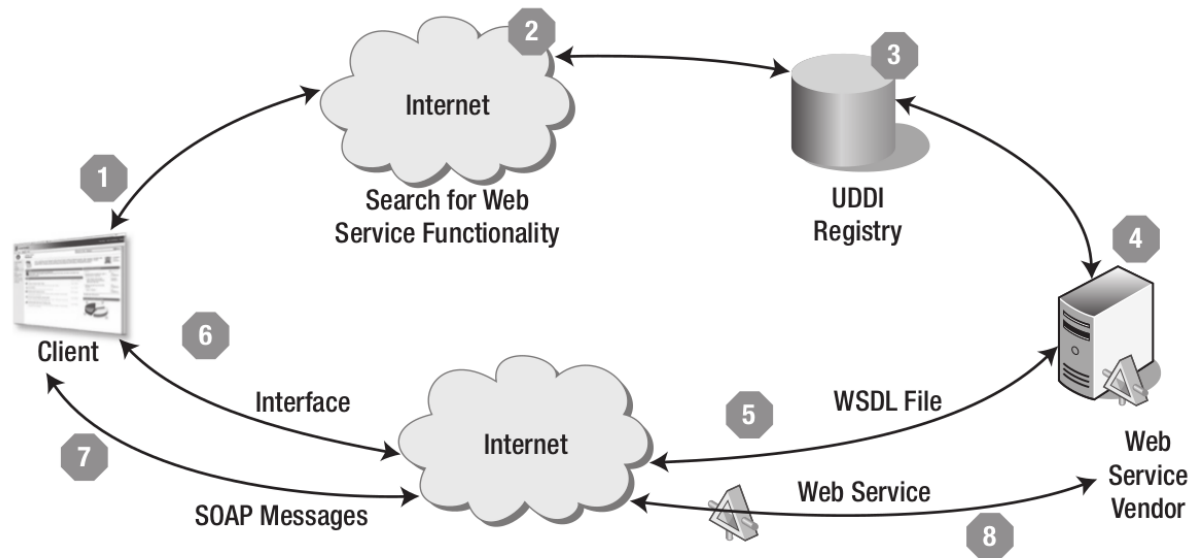
### 2.4.2 REST

REST, *Representational State Transfer*, ou Transferência de Estado Representativo, é um padrão arquitetural para aplicações web definido por Roy Fielding (FIELDING, 2000).

---

<sup>1</sup> <<http://www.w3.org/TR/xml/>>

Figura 3 – Ciclo de Vida de um Serviço Web SOAP



Fonte: (BAHREE *et al.*, 2007)

Segundo o autor, o REST ignora os detalhes de sintaxe de protocolo e de implementação, focando nas regras de negócio, na comunicação entre os componentes e suas restrições. Suas características básicas são a sua essência cliente-servidor, a comunicação sem estado, utilização de cache, interface uniforme e a utilização de camadas.

Assim como no SOAP, as aplicações REST são independentes de plataforma e de linguagem e rodam em cima do protocolo HTTP. No entanto, não há um formato específico para a representação dos dados, sendo possível utilizar tanto o XML, que tem uma estrutura mais pesada, quanto JSON, que é mais leve (SAUDATE, 2014).

Diante do exposto, (RICHARDSON; RUBY, 2008) propõe a utilização destas aplicações como serviços web, o que foi chamado de web services RESTful. Estes serviços se utilizam da infraestrutura disponível na web, através do protocolo HTTP, e devem atender às propriedades abaixo:

- Todos os recursos devem ser endereçáveis por uma URI (*Uniform Resource Identifier*) com nomes significativos;
- Os serviços não devem guardar estado, ou seja, todas as informações necessárias para o processamento devem estar contidas na requisição;
- Deve ter uma interface uniforme e utilizar os verbos HTTP;
- Deve prover links que permitam a navegação por outros serviços correlacionados;

No que se trata da uniformização da interface, como o autor cita, devem ser utilizados

os métodos GET, PUT, POST, DELETE, HEAD e OPTIONS, do protocolo HTTP. É possível atender a todas as operações de um CRUD (criar, ler, atualizar e deletar) utilizando os verbos da seguinte forma: GET para recuperar um recurso, PUT para criar um novo recurso, POST atualizar um recurso e DELETE deletar um recurso. Os demais métodos devem ser utilizados para recuperar metadados, no caso do HEAD, ou para identificar a que chamadas um recurso suporta, com o OPTIONS.

De acordo com (WAGH; THOOL, 2012), estas características dão vantagens ao REST no que diz respeito a um tráfego menor de dados, flexibilidade e maior simplicidade no desenvolvimento dos serviços, no entanto, segundo o autor, o SOAP é mais seguro, por oferecer um protocolo específico para este fim, chamado de WS-Security.

## 2.5 CONSIDERAÇÕES DO CAPÍTULO

Neste capítulo, apresentou-se o referencial teórico utilizado para a realização da pesquisa. Foram apresentados os conceitos de computação ubíqua, onde os ambientes em si são dotados de poder de processamento e a interação humano computador se dá de maneira transparente. Neste contexto, inseriu-se a educação ubíqua, que busca vencer as barreiras físicas do processo de aprendizagem, onde a educação é acessível em todo lugar e a qualquer momento.

Como um meio para se atingir a educação ubíqua, foi apresentada a Internet das Coisas. Tal paradigma trata de coisas, objetos, interagindo e cooperando para atingirem a um objetivo comum e traz uma série de desafios como, o armazenamento, processamento e exibição da massiva quantidade de dados gerados, a heterogeneidade de dispositivos envolvidos e os aspectos de segurança.

Por fim, com o objetivo de solucionar alguns desafios decorrentes dos sistemas de Internet das Coisas, é apresentada a Arquitetura Orientada a Serviços e suas duas principais tecnologias, SOAP e REST. A primeira, mais complexa, utiliza a linguagem XML e utiliza, basicamente, três protocolos para entregar os seus serviços, o SOAP, WSDL e UDDI. Os serviços que fazem o uso do padrão arquitetura REST, utilizam o protocolo a HTTP, através de seus verbos e mensagens de retorno, e não são obrigados a implementar um padrão de formatação de dados específicos, sendo geralmente utilizando o JSON ou XML.

Desta forma, para o desenvolvimento do sistema de Internet das Coisas para gravação de áudios de aula, foi utilizada uma arquitetura orientada a serviços, que implementa o padrão arquitetural REST e fornece uma variedade de serviços através do protocolo HTTP, utilizando

a linguagem JSON para a estruturação dos dados. No próximo capítulo, são apresentados os trabalhos relacionados ao desenvolvimento desta pesquisa.



### 3 TRABALHOS RELACIONADOS

Na literatura encontram-se trabalhos com diversas arquiteturas de sistemas para internet das coisas. São desenvolvidos estudos sobre os mais diversos ramos de aplicação, que vão desde a automação até a informática na educação. Neste capítulo apresenta-se os trabalhos que mais se relacionam com a metodologia proposta por esta pesquisa, bem como possíveis contribuições.

(HEO *et al.*, 2015) propõe uma implementação leve de uma plataforma para Internet das Coisas que utilize virtualização, API *RESTful*, bancos de dados, visualização, *mashup* e sensibilidade ao contexto. Nesta pesquisa, foi desenvolvido um ambiente com o hardware Arduino que realiza o controle de sensores e atuadores e se comunica com o servidor através de wifi ou rede cabeada. Para a atuação baseada em contexto, foram adicionadas regras para a ativação dos aquecedores de uma residência de acordo com o histórico das temperaturas interna e externa recebidas dos sensores.

Os autores desenvolveram uma aplicação onde os sensores e atuadores reais foram virtualizados como ícones em uma página Web, de forma que o usuário pudesse controlá-los através de cliques na tela. Além disto, foram criados gráficos de histórico das medições dos sensores, onde é possível, por exemplo, acompanhar as variações de temperatura do ambiente e, automaticamente, o sistema decidir pelo acionamento ou não dos aquecedores, através de atuadores.

Para criar uma camada única de aplicação entre diversos dispositivos heterogêneos (KOVATSCHE; MAYER; OSTERMAIER, 2012) propuseram uma arquitetura onde infraestrutura e aplicações são independentes e toda a lógica de negócio é executada em um servidor. Nesta abordagem, os dispositivos de internet das coisas consomem recursos do servidor de aplicação utilizando serviços web *RESTful*, o que diminui a complexidade da programação dos dispositivos e centraliza o desenvolvimento das aplicações.

Dentre os objetivos da pesquisa de (KOVATSCHE; MAYER; OSTERMAIER, 2012) estão uma integração completa dos sistemas embarcados com a Web, o desenvolvimento de APIs intuitivas e uniformes, que permitam a integração de diferentes serviços, a separação entre infraestrutura e lógica de programação, tirando a complexidade dos softwares embarcados, e permitir ao usuário adicionar novas funcionalidades aos seus dispositivos sem a necessidade de alteração do *firmware*. Os autores deixam claro que a segurança é um aspecto crucial para este tipo de sistema, no entanto, não fez parte do escopo desta pesquisa.

(GUINARD; TRIFA; WILDE, 2010) trazem em sua pesquisa um abordagem onde os dispositivos físicos, e os dados obtidos por eles, passam a ser parte integrada da Web. O objetivo deste trabalho é tornar mais simples o desenvolvimento de aplicações que cruzam dados de diversas fontes do mundo real. Os autores apresentam uma arquitetura baseada em serviços *RESTful* e demonstram através de protótipos de aplicações a validade da proposta.

Para (GUINARD; TRIFA; WILDE, 2010), o padrão arquitetura REST é a solução mais efetiva para a Web das Coisas, por conta de suas vantagens para sistemas descentralizados e de uso massivo. Para dispositivos que não tenham a capacidade de se conectar diretamente à internet, ou que não implementem o protocolo HTTP, os autores propõem a utilização de um *proxy* reverso, com a função de traduzir as requisições HTTP para os protocolos específicos dos dispositivos.

Em sua pesquisa, (PINTUS; CARBONI; PIRAS, 2012) apresentam o protótipo de uma arquitetura escalável para um sistema de larga escala para a Web das Coisas. Esta plataforma, Paraimpu, permite que pessoas conectem, usem ou compartilhem objetos do mundo real ou virtual, serviços e dispositivos com o objetivo de criar aplicações pervasivas personalizadas. Para garantir a escalabilidade é utilizado o servidor de aplicações Tornado, o balanceador de carga NGINX e o banco de dados não relacional MongoDB.

Segundo os autores, desenvolver uma plataforma social para a Web das coisas traz desafios no sentido de criar modelos e abstrações que possam representar a grande quantidade de objetos que podem vir a fazer parte da solução. (PINTUS; CARBONI; PIRAS, 2012) concluem dizendo que o Paraimpu é uma plataforma social para a Web das Coisas que facilita a adição, compartilhamento, composição, filtragem e adaptação de dados vindos de serviços online, redes sociais e dispositivos (coisas) com acesso à Web.

No trabalho desenvolvido por (XUE; WANG; CHEN, 2011) preocupa-se com o suporte à aprendizagem ubíqua, onde os alunos tenham acesso a conteúdo em todos os lugares e a todo tempo. Para atingir a este objetivo, é proposto um ambiente de aprendizagem ubíquo baseada na internet das coisas. Tal ambiente conta com uma arquitetura em camadas, onde divide-se a camada de percepção (sensores, atuadores, celulares), a de transmissão de dados (rede) e a da aplicação em si.

Para (XUE; WANG; CHEN, 2011), a Internet das Coisas faz com que o mundo real e o virtual estejam conectados, promovendo uma integração perfeita, de forma que o estudante aprende o que ele precisa a todo momento, em todo lugar e com o conteúdo otimizado para a situação, fazendo o ambiente de aprendizagem inteligente. Os autores consideram que a

Tabela 1 – Comparação entre o E-things e os trabalhos relacionados

	<b>Trab. 1</b>	<b>Trab. 2</b>	<b>Trab. 3</b>	<b>Trab. 4</b>	<b>Trab. 5</b>	<b>Proposta</b>
<b>Interoperab.</b>	Sim	Sim	Sim	Sim	Sim	Sim
<b>Alta Disponib.</b>	Não	Não	Não	Sim	Não	Sim
<b>Toler. a Falhas</b>	Não	Não	Não	Sim	Não	Sim
<b>Escalabilidade</b>	Não	Não	Não	Sim	Não	Sim
<b>RESTful API</b>	Sim	Sim	Sim	Sim	Não	Sim
<b>Ensino</b>	Não	Não	Não	Não	Sim	Sim

utilização de Internet das Coisas em ambientes de aprendizagem ubíquos é uma das áreas de pesquisas mais desafiantes.

A fins de comparação, a Tabela 1 traz as principais características do E-things cruzadas com os trabalhos relacionados. Para simplificar a tabela as pesquisas são identificadas da seguinte forma: Trabalho1 – (HEO *et al.*, 2015) , Trabalho2 – (KOVATSCHEK; MAYER; OSTERMAIER, 2012), Trabalho3 – (GUINARD; TRIFA; WILDE, 2010), Trabalho4 – (PINTUS; CARBONI; PIRAS, 2012) e Trabalho5 – (XUE; WANG; CHEN, 2011).

Diante do exposto, esta pesquisa propõe ampliar as contribuições dos trabalhos correlatos por meio da aplicação prática de sistemas de Internet das Coisas no âmbito da educação. Traz-se o desenvolvimento de um cliente para gravação dos áudios de aulas, e registro de comentários, projetado com uma arquitetura orientada a serviços baseada em *web services* RESTful, escalável, interoperável e que garante a alta disponibilidade e tolerância a falhas. Trabalha-se para que este seja um ambiente de imersão, onde os indivíduos estejam cercados pelo conhecimento, e que esta plataforma seja um meio para viabilização de uma educação ubíqua.

### 3.1 CONSIDERAÇÕES DO CAPÍTULO

Neste capítulo, foram apresentados os trabalhos relacionados ao objetivo desta dissertação. Foram selecionadas pesquisas que abordam o desenvolvimento de sistemas, e arquiteturas de sistemas, para a Internet das Coisas, e trazem estudos sobre como enfrentar os desafios intrínsecos estes ambientes, tais quais a massiva geração de dados, a heterogeneidade de dispositivos, a necessidade de estar sempre disponível e a capacidade de atender a grandes quantidades de requisições.

Desta forma, através da ampliação das contribuições dos trabalhos apresentados pretende-se atingir ao objetivo desta pesquisa, de propor um sistema computacional que utilize dispositivos de Internet das Coisas como ferramenta de suporte a professores e alunos com a

gravação de áudios de aulas. No próximo capítulo será apresentada a metodologia utilizada na pesquisa, bem como os recursos tecnológicos e os procedimentos para análises dos dados.

## 4 METODOLOGIA

A operacionalização desta pesquisa foi realizada na Universidade Estadual do Maranhão – UEMA, por meio do Centro de Ciências Tecnológicas – CCT. Dentre os cursos oferecidos por este centro, foi selecionado o Curso de Engenharia da Computação, em virtude da sua proximidade com o Programa de Pós-Graduação em Engenharia da Computação e Sistemas.

Esta pesquisa tem como o objetivo geral propor um sistema computacional que utilize dispositivos de Internet das Coisas como ferramenta de suporte a professores e alunos com a gravação de áudios de aulas. Para atingir a este objetivo, dividem-se os esforços nos seguintes objetivos específicos: 1) Realizar levantamento bibliográfico das principais características e requisitos de sistemas de Internet das Coisas; 2) Identificar e analisar os fundamentos da área de computação ubíqua, em especial a sua aplicação em ambientes educacionais; 3) Apresentar a arquitetura da plataforma proposta, que dê suporte ao desenvolvimento de aplicações de Internet das Coisas para a educação; 4) Levantar os requisitos e propor sistema para gravação e disponibilização online de áudios de aulas, bem como registros de anotações, utilizando dispositivo de Internet das Coisas; 5) Apresentar protótipo e especificações do sistema para gravação de aulas e registro de anotações; 6) Disponibilizar a alunos, em ambiente controlado, acesso ao protótipo desenvolvido do sistema; 7) Verificar, do ponto de vista técnico, se os requisitos elencados são atendidos pelo protótipo durante a utilização dos usuários; 8) Analisar a satisfação dos usuários em relação ao sistema proposto do ponto de vista dos áudios gravados e anotações realizadas;

Os resultados da pesquisa, quanto aos dois primeiros objetivos específicos, são relatados no Capítulo 2; quanto ao terceiro, quarto e quinto, são descritos no Capítulo 5; o sexto, sétimo e oitavo são apresentados nos Capítulo 6 e Capítulo 7 desta dissertação. Por se tratar de um número limitado de alunos pesquisados, as análises desses resultados foram apenas qualitativas.

### 4.1 SUJEITOS DA PESQUISA

Considerando o curso de Engenharia de Computação da UEMA, trabalhou-se com alunos da graduação, sendo cadastrados, no sistema proposto, 43 (quarenta e três) usuários, que utilizaram as ferramentas de envio de anotações e reprodução dos áudios gravados, além de avaliarem o perfeito funcionamento do sistema e a sua efetividade como ferramenta para elucidação de dúvidas e revisão de conteúdos.

A escolha destas disciplinas se deu devido o envolvimento dos alunos e professores com a proposta apresentada para o desenvolvimento da dissertação nesta pesquisa. A prática adotou ainda a elaboração de instrumentos para coletar dados, a preparação de um ambiente de servidores, a instalação do protótipo desenvolvido nestes servidores, a configuração de turmas específicas para cada disciplina, a configuração do cliente Raspberry e a concessão dos devidos acessos aos usuários.

#### 4.2 INSTRUMENTO DE COLETA DE DADOS

Nesta etapa, utilizou-se a escala de Likert como instrumento de coleta de dados, considerando sua natureza autoadministrativa, onde o indivíduo informa suas próprias respostas. Nesta escala, utilizam-se conjuntos de afirmações as quais são solicitadas ao indivíduo manifestar a sua reação na forma de pontos, em graus de concordância ou discordância (SAMPIERI; COLLADO; LUCIO, 2003).

Foram utilizados questionários com assertivas onde os alunos registraram a sua concordância através de cinco graus da escala Likert. O objetivo destes questionários é qualificar e avaliar o sistema proposto de forma que, os graus da escala foram apresentados da seguinte forma: (1) Discorda plenamente, (2) Discorda, (3) Nem concorda nem discorda, (4) Concorda e (5) Concorda plenamente.

Desta forma, realizou-se a elaboração dos questionários com questões abertas e fechadas, conforme Apêndice A. As questões fechadas contêm alternativas de resposta previamente delimitadas. Enquanto, as questões abertas, não se delimitam às alternativas de resposta.

#### 4.3 PROCEDIMENTOS PARA COLETA DE DADOS

No primeiro semestre de 2017, os alunos foram informados sobre os objetivos específicos da pesquisa e iniciou-se o cadastramento no ambiente disponibilizado dos participantes que fizeram parte da pesquisa. Para avaliar o sistema, os participantes receberam uma breve explicação das funcionalidades existentes e foram orientados à utilizar o software em seus próprios *smatphones* ou computadores, durante uma breve aula sobre Internet das Coisas.

Para a validação desta pesquisa, foi realizado um estudo por amostra, composto por 43 (quarenta e três) alunos. Os dados foram coletados a partir de questionários aplicados utilizando-se o Google Forms, onde aos alunos foi submetido o formulário presente no Apêndice A.

#### 4.4 RECURSOS TECNOLÓGICOS

Para o desenvolvimento desta pesquisa, a proposta de um sistema computacional que utilize dispositivos de Internet das Coisas como ferramenta de suporte a professores e alunos com a gravação de áudios de aulas, se fez necessário a utilização de diversos recursos tecnológicos, a fim de garantir a alta disponibilidade, tolerância a falhas, balanceamento de carga, escalabilidade e heterogeneidade de dispositivos. Foram utilizados uma arquitetura orientada a serviços, com *web services* REST, ambiente nodeJS, banco de dados MongoDB, balanceador e servidor Web Nginx, Bootstrap CSS, HTML, JavaScript, PHP e um dispositivo Raspberry Pi e seus periféricos (microfone, fonte de alimentação e *hub* USB).

#### 4.5 CONSIDERAÇÕES DO CAPÍTULO

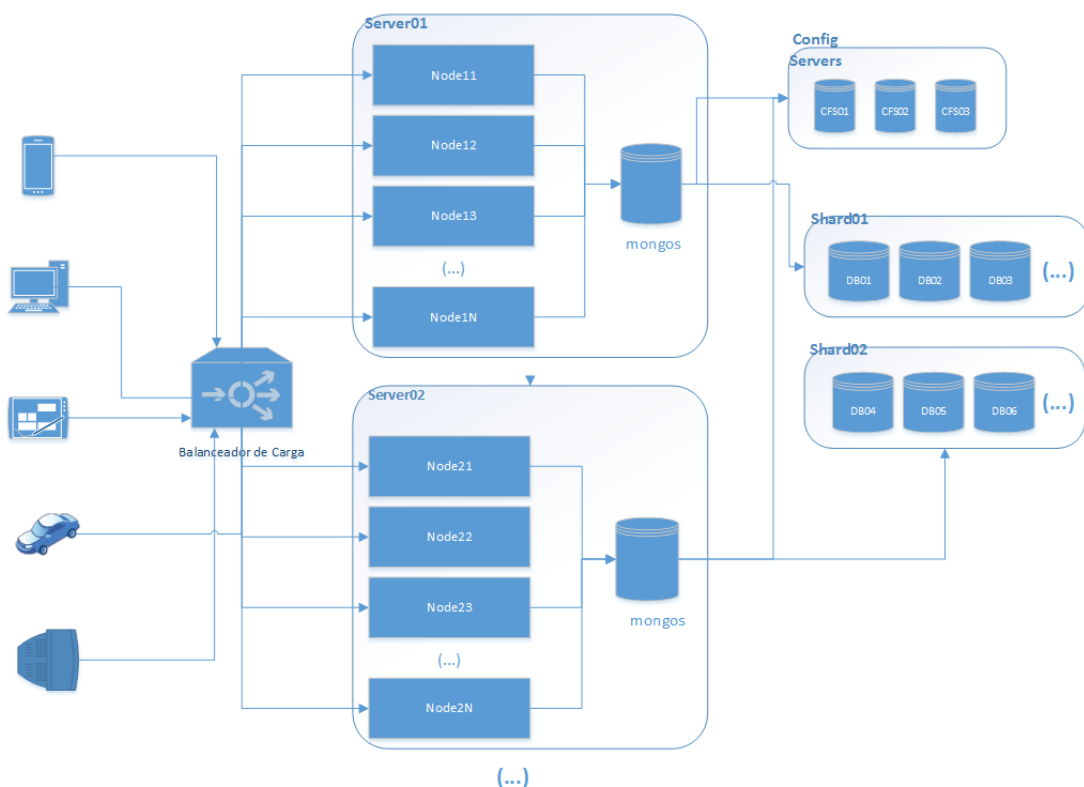
Foram apresentados neste capítulo a estrutura de tópicos da dissertação e a metodologia utilizada na pesquisa, bem como, os sujeitos da pesquisa, os instrumentos e procedimentos de coleta e análise de dados e os recursos tecnológicos utilizados do sistema de Internet das Coisas para gravação de áudios de aulas. No próximo capítulo é trazida a arquitetura da plataforma proposta, que dê suporte ao desenvolvimento de aplicações de Internet das Coisas para a educação, o levantamento de requisitos para o sistema de gravação, disponibilização online de áudios de aulas e registros de anotações e é apresentado o protótipo do sistema desenvolvido.

## 5 PROTOTIPAGEM DO SISTEMA PROPOSTO

### 5.1 ARQUITETURA DA PLATAFORMA PROPOSTA

Este trabalho tem como um de seus objetivos propor uma plataforma educacional escalável e redundante para a internet das coisas, de modo a atender todos os desafios que este tipo de ambiente proporciona. Desta forma, este capítulo apresenta o e-things, uma plataforma que se propõe a fornecer tecnologias educacionais para dispositivos da internet das coisas através de serviços web, que conta com uma arquitetura de servidores e bancos de dados que propiciam alta disponibilidade e tolerância a falhas. O desenho completo desta arquitetura pode ser observado na Figura 4.

Figura 4 – Arquitetura Proposta



Fonte: Elaborado pelo autor

Este capítulo se divide da seguinte forma: requisitos básicos do sistema, que elencará as funcionalidades indispensáveis para o funcionamento do software; arquitetura de software, que explicará aspectos técnicos do sistema; ambiente de banco de dados, que trará as definições de tecnologias e topologia da base de dados; e infraestrutura de servidores, que apresentará a organização dos servidores de aplicação bem como o seu relacionamento com o balanceador de

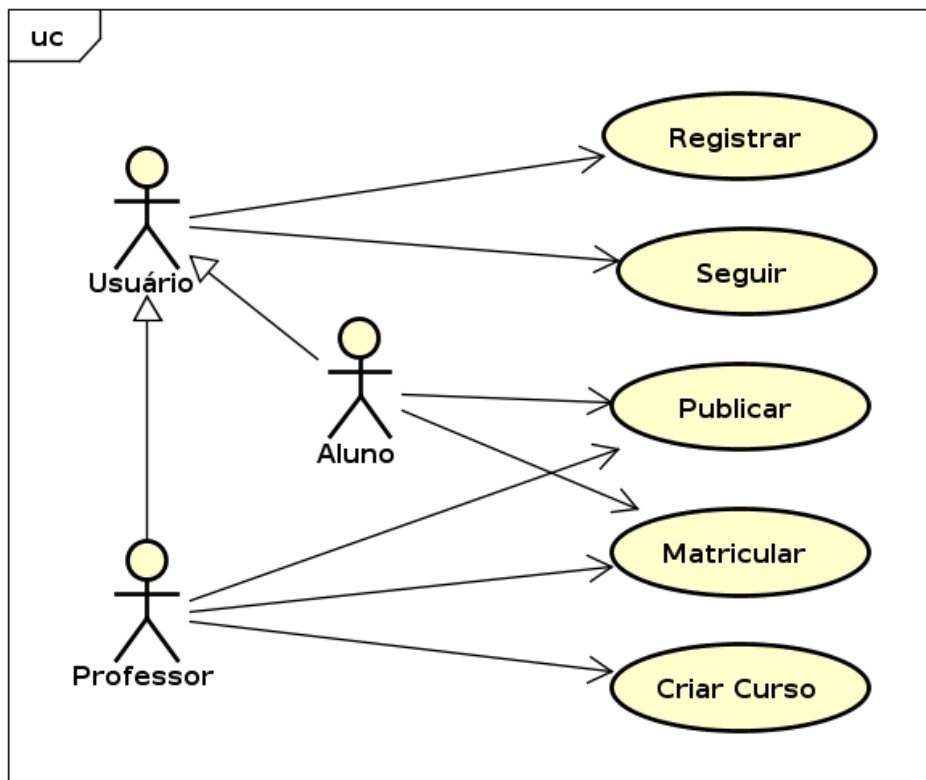


carga e o banco de dados, explicando como tem-se garantidas as características propostas.

### 5.1.1 Requisitos Básicos do Sistema

Para a definição de uma plataforma educacional específica para a Internet das Coisas, faz-se necessária a elucidação e o levantamento dos requisitos de software mínimos deste sistema, levando-se em consideração todas as características deste tipo de ambiente. Esta pesquisa traz uma série de requisitos básicos que acredita-se ser o mínimo para um software desta natureza. A plataforma proposta é aberta para extensão, de modo que, o desenvolvimento de módulos específicos, como um cliente para gravação de aulas, pode vir a demandar novas funcionalidades e o desenvolvimento de novos serviços. A Figura 5 apresenta o diagrama de caso de uso destes requisitos.

Figura 5 – Diagrama de Casos de Uso Básicos



Fonte: Elaborado pelo autor

Como característica geral, é importante que o sistema apresentado seja acessível e disponível a qualquer pessoa. Que qualquer usuário possa se registrar e ter acesso às turmas criadas, que possa se matricular e passar a contribuir com os demais alunos. Faz-se necessário

também que todos os usuários registrados possam criar os seus próprios cursos, tornando-se assim professores, e que o sistema saiba diferenciar estes dois papéis, o mesmo usuário como aluno e como professor.

Outro requisito essencial para o sucesso da proposta, é a capacidade de armazenar diferentes tipos de conteúdo. Pelo fato da arquitetura ser extensível ao desenvolvimento de novos módulos, é necessário que o sistema suporte os novos tipos de dados e de conteúdo gerados por estas funcionalidades, que disponha de um mecanismo de postagens genérico, sem estrutura fixa e que se adéque a diversos formatos de dados.

Por fim, aproveitando o sucesso das redes sociais, deve ser possível desenvolver uma linha do tempo, onde usuários possam seguir outros usuários ou cursos, e ter acesso, em seu perfil, as principais atualizações das pessoas que segue, bem como recursos criados, postagens em fóruns, anotações de aulas e etc.

### **5.1.2 Arquitetura do Software**

Diante a pluralidade de dispositivos (coisas) que compõe a IoT, é necessário utilizar um padrão arquitetural interoperável, independente de plataforma e leve, para o desenvolvimento da solução. Desta forma, a arquitetura orientada serviços, associada à utilização de *web services RESTful*, atende bem a este propósito.

Sendo assim, a plataforma oferece um conjunto de serviços REST, aderentes às características propostas por (FIELDING, 2000) e (RICHARDSON; RUBY, 2008). Estes serviços compõe uma API (*Application Programming Interface*), que utiliza a infraestrutura cliente-servidor do protocolo HTTP, seus verbos e mensagens de retorno, para atender aos requisitos do sistema.

Padronizou-se que os verbos HTTP seja utilizados da seguinte forma: GET para recuperar informação, POST para salvar um novo registro, PUT para editar algum dado e DELETE para remover alguma entrada. Além disto, as mensagens de retorno do protocolo e seus códigos de estado são utilizados como informação adicional nas respostas dadas pela API, de forma que, para cada situação, um código diferente é retornado. Na Figura 6 tem-se um exemplo de uma chamada ao serviço de consulta de usuário. A lista completa com as especificações dos *end-points* e mensagens de retorno pode ser encontrada no Apêndice B.

As questões referentes à segurança na plataforma se dão em dois níveis, o da confidencialidade dos dados trafegados na rede e o da autorização e autenticação. O primeiro é

Figura 6 – Exemplo de Chamada e Retorno da API

```

GET /api/user/thiago HTTP/1.1
Host: localhost:3000
Connection: keep-alive
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/46.0.2490.86 Safari/537.36
Cache-Control: no-cache
Authorization: eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJfcmFwaWQiOiIiNDY1MDViMDFmYTAzY
Accept-Encoding: gzip, deflate, sdch
Accept-Language: pt-BR,pt;q=0.8

HTTP/1.1 200 OK
X-Powered-By: Express
Content-Type: application/json; charset=utf-8
Content-Length: 144
ETag: W/"90-/cnyFFsRLfpEG7Py403RfA"
Date: Sat, 28 Nov 2015 17:54:12 GMT
Connection: keep-alive

[{"@type": "d", "username": "thiago", "name": "Thiago Nunes de Sousa", "email": "thiagonunes.tns@gmail.com", "rid": "#13:0", "@rid": "#-2:0", "@version": 0}]

```

Fonte: Elaborado pelo autor

resolvido através da utilização de servidor com suporte a TLS (*Transport Layer Security*), que garante a confidencialidade através da criptografia das mensagens. No que tange à autenticação e autorização, a API utiliza um mecanismo baseado em *tokens*. Com este mecanismo, após a realização do login, através do *end-point* auth, é enviada uma chave de acesso ao cliente. Esta chave passa a ser enviada em todas as requisições subsequentes no cabeçalho *Authorization*, e, com isso, o acesso aos serviços é autorizado. A Figura 7 ilustra este fluxo, com a chamada ao método de *login* e a passagem do *token* na chamadas seguintes.

Por fim, para lidar com a quantidade massiva de requisições que este tipo de sistema está exposto, utiliza-se o Node.js<sup>1</sup>, um *framework* assíncrono, orientado a eventos, que permite a construção de aplicações escaláveis. E, por conta de possíveis limitações de conectividade dos clientes, e com o objetivo de diminuir ao máximo o tráfego de informações, a API estrutura seus dados utilizando JSON<sup>2</sup> (*JavaScript Object Notation*), uma linguagem leve, baseada em texto, para intercâmbio de informações, que define poucas regras de formatação para a estruturação dos dados (CROCKFORD, 2006).

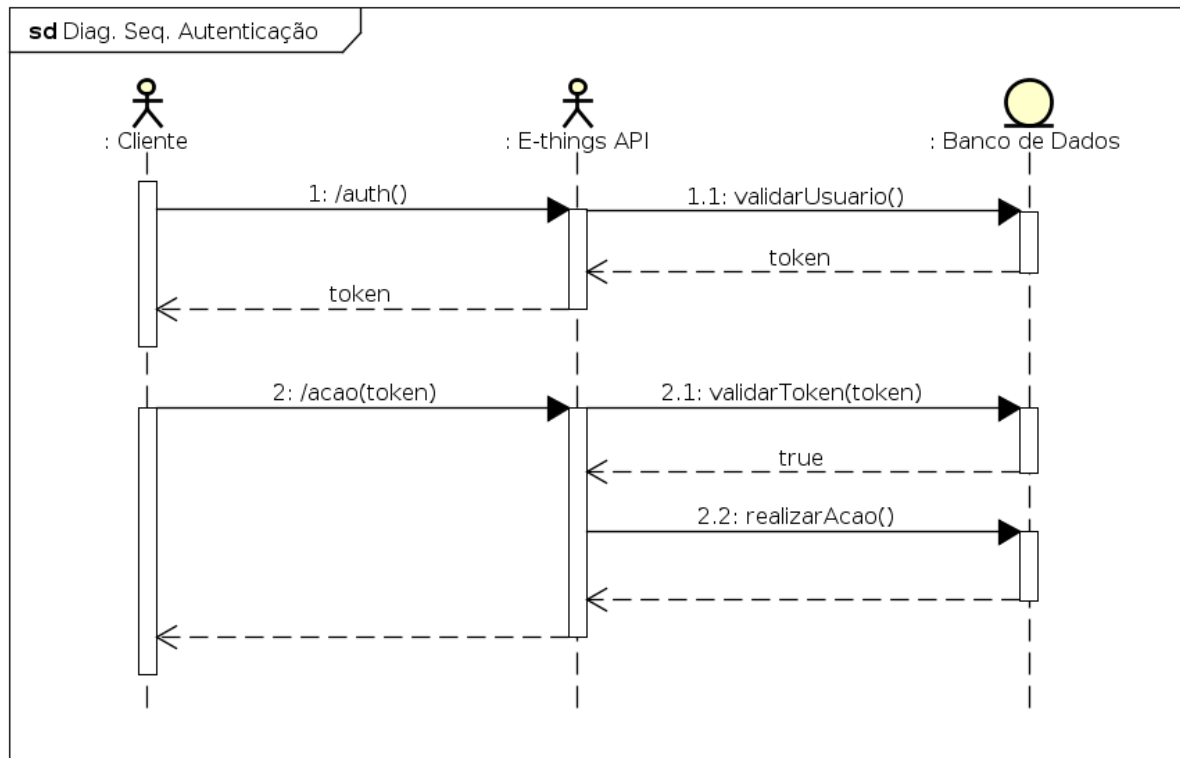
### 5.1.3 Ambiente de Banco de Dados

Considerando o requisito de adaptabilidade a diferentes formatos de dados, a plataforma proposta utiliza um modelo de banco de dados orientado a documentos. Este tipo de banco

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

<sup>2</sup> <<http://www.json.org/>>

Figura 7 – Diagrama de Sequência do mecanismo de autenticação



Fonte: Elaborado pelo autor

permite que registros sejam inseridos sem um esquema pré-definido (*schemaless*), como ocorre nos SGBD's tradicionais, dando mais flexibilidade à base. Além disto, o armazenamento dos dados se dá na forma de documentos escritos em XML, JSON, ou outras linguagens (NAYAK; PORIYA; POOJARY, 2013).

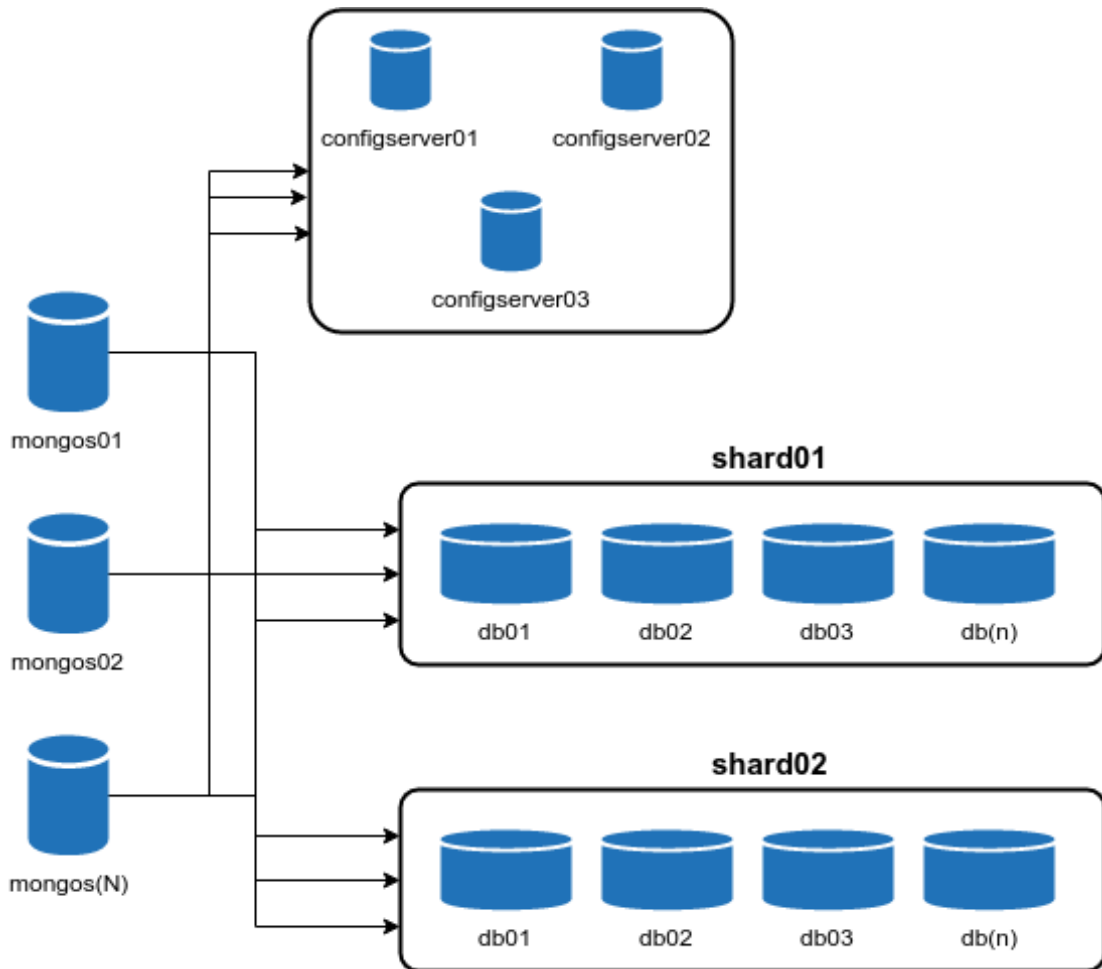
Desta forma, o software desenvolvido utiliza o MongoDB<sup>3</sup> como base de dados. Este banco se encaixa na categoria dos bancos NoSQL (*Not Only SQL*), sendo orientado a documentos, com suporte a replicação e particionamento de dados (CHODOROW, 2013). Além disto, dá suporte a dados binários, conta com uma poderosa linguagem de consultas e garante alta performance ao se trabalhar com bases muito grandes (HAN *et al.*, 2011).

Para garantir a redundância dos dados e minimizar qualquer problema de perda de informações, será utilizado o mecanismo de *replicaset* do MongoDB. Desta forma, registros inseridos na base são replicados em servidores distintos, garantindo a acessibilidade aos dados mesmo com a indisponibilidade de alguma das instâncias.

De maneira complementar ao mecanismo de réplicas, é utilizado também o particionamento de dados (*sharding*). Esta funcionalidade permite que registros de tipos diferentes

<sup>3</sup> <<https://www.mongodb.com/community>>

Figura 8 – Topologia do ambiente de Banco de Dados



Fonte: Elaborado pelo autor

possam ser armazenados em servidores distintos, possibilitando, por exemplo, a segregação de dados por área de interesse ou localização geográfica. Esta funcionalidade incrementa a performance das consultas uma vez que, quando se trata de Internet das Coisas, há a expectativa da massivas geração de dados. A topologia de servidores de bancos de dados com *shards* e *replicasets* pode ser observada na Figura 8.

A organização interna dos dados se dá em três coleções: usuários, cursos e recursos. Os cursos são agrupamentos de recursos, criados por um usuário (professor) para outros usuários (alunos). A coleção recursos armazena todo conteúdo produzido na plataforma, desde aulas, respostas em fóruns, anotações, postagens, vídeos e etc. Esta coleção, com a característica *schemaless* do banco de dados, permite o desenvolvimento do modelo genérico de postagens, pois aceita registros em qualquer formato, cabendo à aplicação diferenciá-los pelo seu tipo.

Com estas características e configurações de bancos de dados, a arquitetura proposta atende a alguns dos principais requisitos dos sistemas para Internet das Coisas, do ponto de vista

dos dados, que são a alta disponibilidade, tolerância a falhas, redundância, escalabilidade e a capacidade de trabalhar com grandes quantidade de dados e em diversos formatos.

#### 5.1.4 Infraestrutura de Servidores

Da mesma maneira que o ambiente de banco de dados, os servidores de aplicação são dispostos de modo a atenderem aos requisitos das aplicações de Internet das Coisas. A configuração da arquitetura permite, através do balanceamento de carga, que o sistema tenha performance compatível com a quantidade de processamento que for necessário.

Como balanceador de carga, utiliza-se o Nginx<sup>4</sup>, que centraliza todas as requisições e as roteia de maneira equilibrada entre os nós do sistema. Além disto, esta configuração oferece a possibilidade de adição de novos nós ao parque de servidores, conforme o crescimento da demanda, garantindo assim a escalabilidade horizontal do ecossistema.

No que diz respeito aos servidores de aplicação, é preciso considerar a natureza *singlethread* do NodeJS. Para vencer esta limitação, é executada uma instância da aplicação para cada núcleo de processador da máquina, de forma que, cada uma destas instâncias é com cópia do mesmo software, a API *RESTful* disponibilizada para os clientes da plataforma.

#### 5.1.5 Considerações sobre a plataforma

Observa-se que, todas as configurações da plataforma proposta, e todas as decisões tomadas nesta etapa da pesquisa, visam atender aos requisitos mínimos para uma aplicação de Internet das Coisas para a área de educação. Os requisitos levantados trazem as funcionalidade mínimas para um software desta natureza. A arquitetura orientada a serviços, com *web services* REST, padroniza um protocolo de comunicação e garante que dispositivos diversos, mesmo que com pequeno potencial de processamento, tenham acesso à aplicação. O banco de dados MongoDB oferece suporte para o mecanismo de postagens genérica, bem como garante performance nas consultas e a redundância dos dados. E, por fim, a arquitetura de servidores com balanceador de carga e aplicações NodeJS, dão à aplicação a capacidade de se adaptar ao crescimento do número de requisições, e garante a alta disponibilidade.

Até este ponto, tem-se uma plataforma genérica para aplicações de Internet das Coisas aplicadas à educação. Tal plataforma é aberta à extensão, e, a criação de aplicações específicas, pode demandar o desenvolvimento de novas funcionalidades, conforme os seus

---

<sup>4</sup> <<https://www.nginx.com/>>

próprios requisitos. No próximo capítulo deste trabalho, traz-se um cliente específico para a plataforma, cujo o objetivo é a gravação de áudios de uma sala de aula. São apresentados os casos de uso específicos deste módulo, bem como os novos *end-points* desenvolvidos, alguns diagramas UML e dispositivos utilizados na solução.

## 5.2 CLIENTE DE GRAVAÇÃO DE AULAS

A partir deste ponto, esta pesquisa passa a detalhar o projeto e o desenvolvimento do cliente para gravação de áudios de sala de aula. Esta aplicação tem o objetivo de, utilizando dispositivos de internet das coisas, realizar a gravação de áudio de aulas e oferecer esta gravação, de maneira automática, em ambiente Web, atuando como ferramenta de suporte a professores e alunos.

Além da gravação de áudio, este módulo oferece aos alunos a possibilidade do registro de anotações e apontamentos, referentes ao assunto da aula, em tempo real. A ferramenta permite que, alunos e professores, tenham acesso, após a aula, às anotações e gravações de áudio a fim de revisar e reforçar o assunto exposto, bem como, eventualmente, tirar dúvidas sobre tópicos tratados na aula.

A solução de gravação completa é constituída por três componentes, a API, um cliente Web, para interação com professores e alunos, e um cliente Raspberry Pi, responsável pela gravação e upload do áudio em si. Neste abaixo são apresentados estes componentes e a sincronia das mensagens trocadas entre os mesmo, bem como os casos de uso do sistema e tecnologias utilizadas no desenvolvimento.

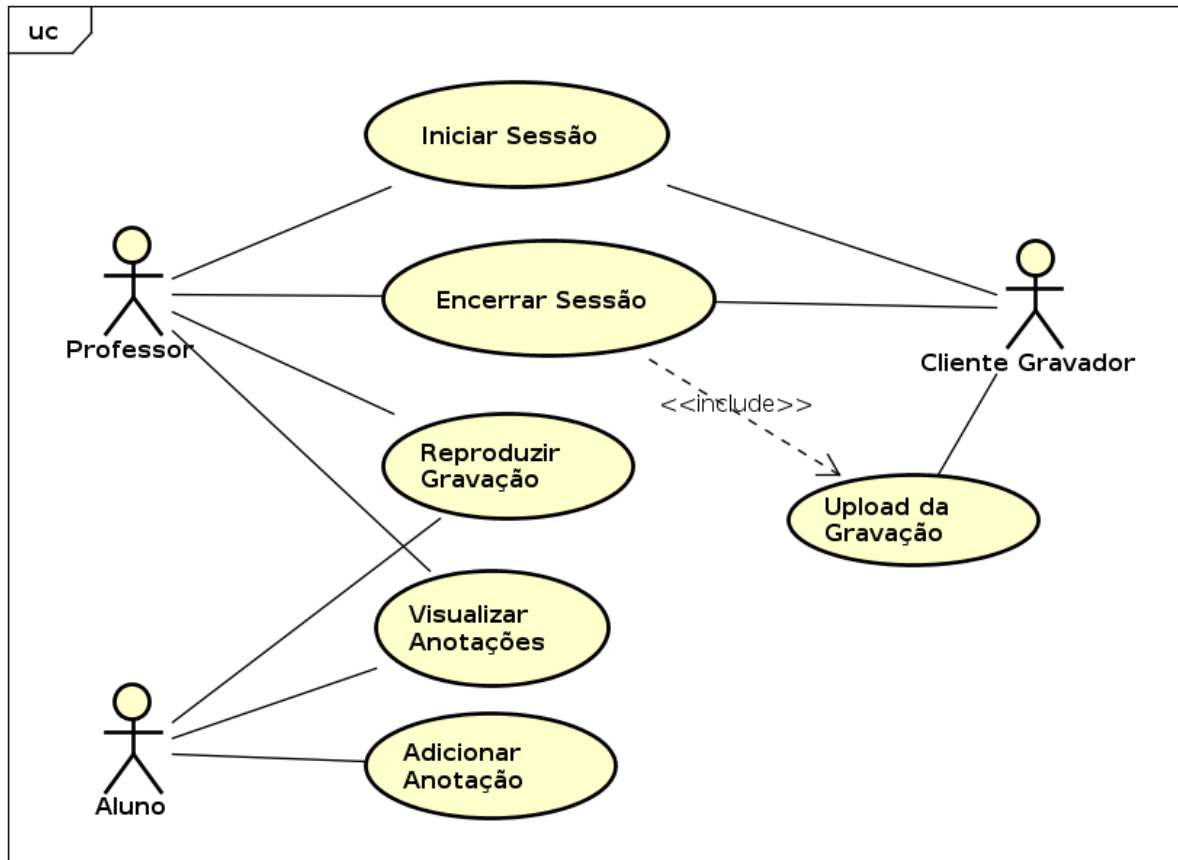
### 5.2.1 Casos de Uso

Além dos casos de uso genéricos implementados pela plataforma, este módulo traz requisitos específicos de suas funcionalidades. São necessárias novas interações como o início ou fim das sessões de gravação e o envio do áudio gravado ao servidor. A Figura 9 traz o diagrama específico com estes casos de uso, que serão discutidos em seguida.

#### 5.2.1.1 Iniciar Sessão

Este caso de uso é o evento inicial de uma gravação, representa o início da aula pelo professor. Esta funcionalidade é acionada pelo professor a partir da interface Web, de modo que, é iniciada a sessão de gravação, e é liberado o acesso dos alunos ao ambiente para registro de

Figura 9 – Diagrama de Casos de Uso



Fonte: Elaborado pelo autor

anotações.

Como o início da sessão, o dispositivo de gravação inicia o processo de captação do áudio através de seu microfone. Esta gravação é realizada até que seja enviada à API um pedido de finalização da sessão.

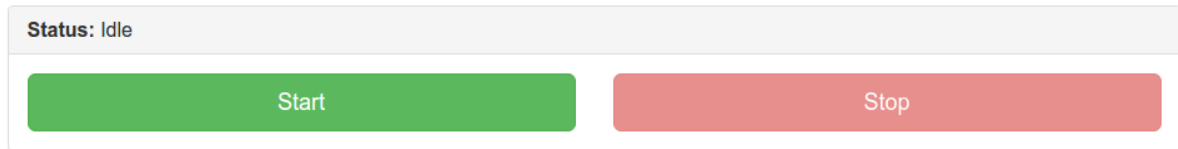
#### 5.2.1.2 Encerrar Sessão

O encerramento da sessão é também feito através da interface Web e só pode ser realizado pelo professor que a iniciou. No entanto, o envio de anotações pelo aluno continua permitido, de forma a estimular a criação de conteúdo e interação entre consumidores daquele recurso.

Com esta ação, o comando de encerramento da gravação é enviado ao Raspberry Pi, que salva o áudio gravado e realiza o envio do arquivo para o servidor, através do caso de uso Upload de Gravação. A interface para acionamento das funcionalidade de início e encerramento



Figura 10 – Interface para Iniciar e Encerrar Sessão



Fonte: Elaborado pelo autor

da sessão pode ser observada na Figura 10.

#### 5.2.1.3 Upload da Gravação

Esta funcionalidade é de uso exclusivo do cliente de gravação e sempre é executada após um encerramento de uma sessão de gravação. Após o recebimento da solicitação de interrupção da gravação, o cliente do Raspberry Pi separa o arquivo contendo o áudio gravado e o envia para o servidor, através da chamada ao serviço de upload da API.

Após o sucesso do envio do arquivo, a API confirma o encerramento da sessão de gravação e libera no cliente Web o áudio da aula, para que o mesmo possa vir a ser acessado pelos alunos.

#### 5.2.1.4 Adicionar, Visualizar Anotação e Reproduzir Áudio

As funcionalidades de adicionar e visualizar anotações são liberadas para os alunos a partir do momento que a sessão é iniciada. Através destes casos de uso, é possível adicionar notas, lembretes, dicas ou qualquer tipo de texto que o aluno julgue relevante para aquele contexto. Já a opção para reproduzir o áudio gravado, só fica disponível após o encerramento da sessão.

A listagem de anotações se dá de forma que todos os usuários têm acesso a todas as anotações, independente de ser o seu criador ou não. Isto permite que o conhecimento, dúvidas ou dicas, de outros alunos possam contribuir com o aprendizado da classe como um todo, tornando o recurso mais rico. A Figura 11 traz a tela com as opções de adição e listagem de anotações e reprodução da gravação.

### 5.2.2 Estados

O fluxo de funcionamento deste módulo de gravação pode ser dividido em estados entre o cliente Web e o Raspberry Pi. Como pode ser observado na Figura 12, o funcionamento do

Figura 11 – Interface de detalhamento da sessão

**Id**  
58588fabc9d9e56fdbcb909bf

**Date**  
19/12/2016 22:55:55

**Audio**  
1:14

**Status**  
ended

**Add Note**

Text

Add

**Note List**

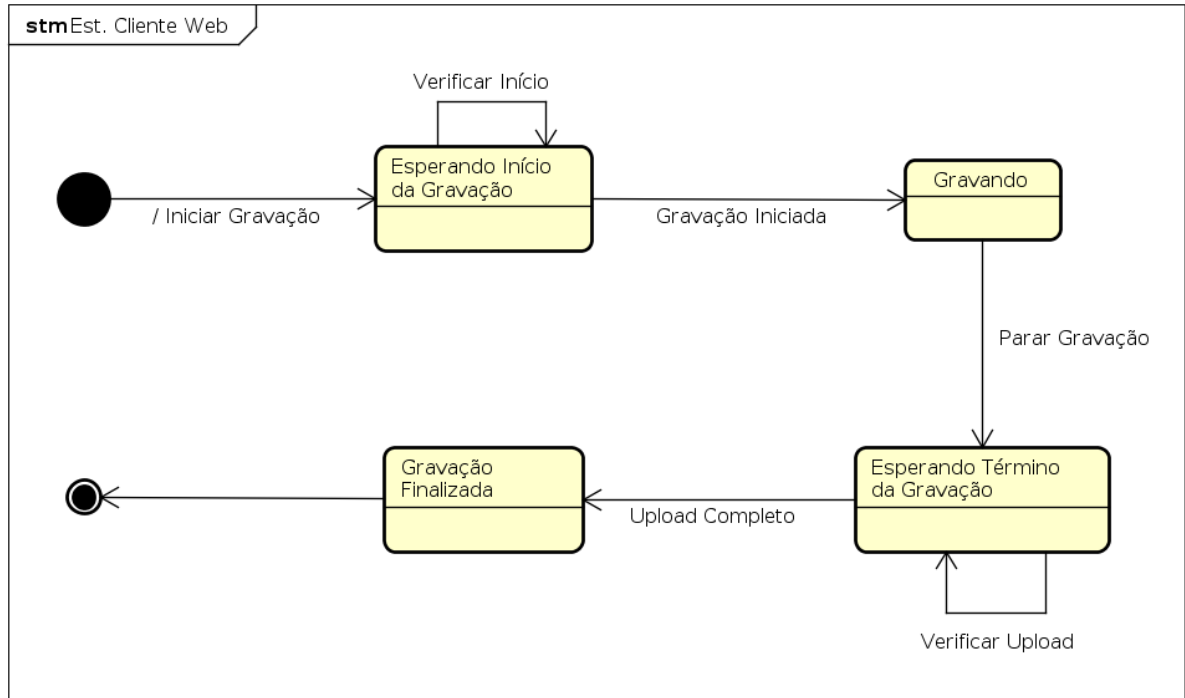
Date/Time	User	Note
19/12/2016 22:57:42	user01	First Note
19/12/2016 22:58:45	user02	Second note
19/12/2016 22:59:28	user02	Other annotation
21/12/2016 21:28:32	user01	After Rec

Fonte: Elaborado pelo autor

cliente Web passa por quatro estados, na seguinte sequência: ao professor iniciar uma sessão de gravação, o sistema fica aguardando a confirmação do início efetivo da gravação pelo Raspberry Pi; ao receber tal confirmação, o sistema passa para o estado gravando, libera a interface para inclusão de comentário, e permanece neste estado até uma solicitação de encerramento da sessão; ao receber o comando de encerramento da sessão, o sistema fica aguardando a efetiva parada do gravador de áudio e o envio do arquivo contendo a gravação; após esta etapa, o áudio gravado fica disponível para reprodução e o ciclo do cliente Web se encerra.

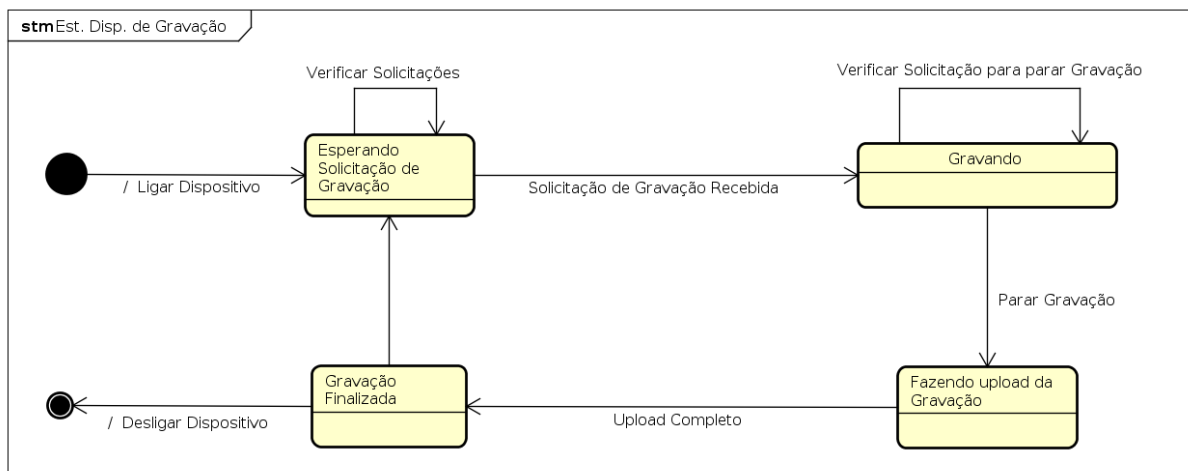
No que diz respeito ao dispositivo de gravação, Raspberry Pi, os seus estados internos são complementares aos do cliente Web, de forma a garantir o perfeito funcionamento do software. A Figura 13 mostra o diagrama com estes estados, e a sequência de interações se dá da seguinte forma: a partir do momento que o dispositivo é ligado, o cliente é iniciado e fica aguardando por solicitações de início de sessão de gravação; ao receber tal comando, o sistema ativa os mecanismo para gravação do áudio do ambiente até que receba instrução para parar; ao receber o comando de encerramento da sessão, o sistema para a gravação e realiza o *upload* do arquivo contendo o áudio; após o envio do arquivo, o sistema volta ao estado inicial, e periodicamente verifica se há novas solicitações de início de sessão.

Figura 12 – Diagrama de Estado do Cliente Web



Fonte: Elaborado pelo autor

Figura 13 – Diagrama de Estado do Dispositivo de Gravação



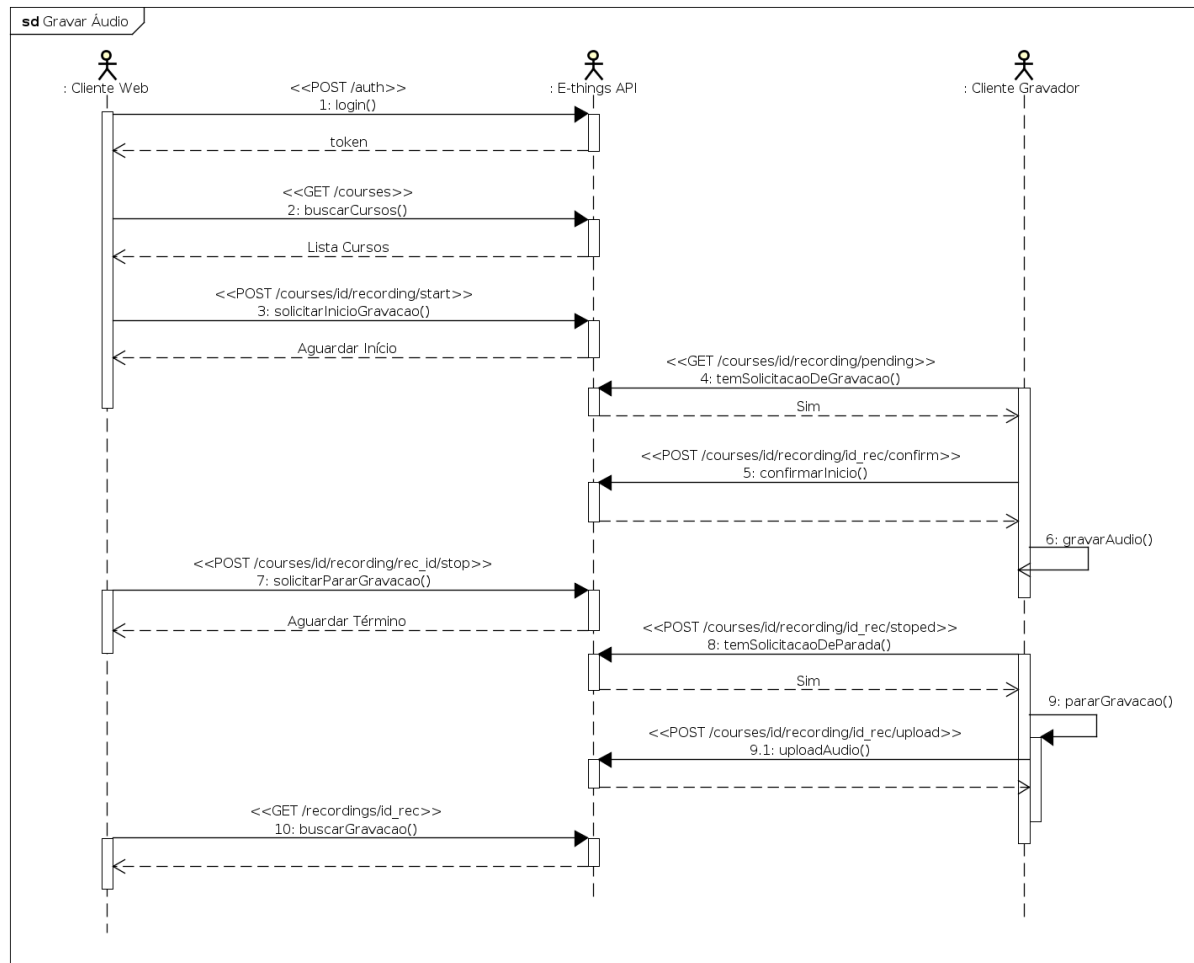
Fonte: Elaborado pelo autor

### 5.2.3 Serviços Web Desenvolvidos

Para a construção do cliente de gravação, fez-se necessário o desenvolvimento de novos *web services*, estendendo-se as funcionalidades já oferecidas pela arquitetura inicial e implementando-se novos recursos. Abaixo tem-se a listagens destes novos *end-points*.

- **POST *courses/:id\_course/recording/start***: Inicia uma sessão de gravação para curso informado; em caso de sucesso retorna o código 201, caso contrário retorna 412 e mensagem explicativa;
- **GET *courses/:id\_course/recording/pending***: Verifica se há alguma sessão de gravação pendente para o curso informado, iniciada pelo cliente Web mas ainda não confirmada pelo Raspberry Pi; caso haja, retorna código 200, caso contrário, 404;
- **PUT *courses/:id\_course/recording/:id\_rec/confirm***: Confirma o início da gravação, utilizado pelo Raspberry Pi; em caso de sucesso retorna código 200, caso contrário retorna 404 ou 412 com mensagem explicativa;
- **GET *courses/:id\_course/recording/running***: Verifica se existem sessões de gravação iniciadas para o curso informado; caso haja, retorna código 200, caso contrário, 404;
- **PUT *courses/:id\_course/recording/:id\_rec/stop***: Encerram uma sessão de gravação; em caso de sucesso retorna código 200, caso contrário retorna 404 ou 412 com mensagem explicativa;
- **GET *courses/:id\_course/recording/:id\_rec/stoped***: Verifica se a sessão de gravação foi encerrada pelo cliente Web; caso tenha sido encerrada, retorna código 200, caso contrário, 404;
- **PUT *courses/:id\_course/recording/:id\_rec/upload***: Realiza o envio do arquivo com o áudio gravado e marca o registro da gravação como finalizado; em caso de sucesso retorna código 200, caso contrário retorna 404 ou 412 com mensagem explicativa;
- **GET *recordings/:id\_rec***: Retorna as informações da gravação, exceto o arquivo com o áudio; em caso de sucesso retorna código 200 e as informações da gravação, caso a gravação não seja encontrada retorna código 404;
- **GET *recordings/:id\_rec/content***: Recupera o áudio gravado para a gravação informada; em caso de sucesso retorna código 200 e o arquivo com a gravação, caso a gravação não seja encontrada retorna código 404, caso o upload ainda não tenha sido realizado, retorna 412;
- **POST *recordings/:id\_rec/note***: Cria uma nova anotação para a anotação informada; em

Figura 14 – Diagrama de Sequência da Gravação de Áudio



Fonte: Elaborado pelo autor

caso de sucesso retorna o código 201, caso contrário retorna 412 e mensagem explicativa;

- **GET /notes?recording=id\_rec:** Lista as anotações referentes à sessão de gravação informada; em caso de sucesso retorna código 200 e as informações das anotações, caso não tenham sido encontrados registros, retorna código 404.

## 5.2.4 Sincronismo das Mensagens

Considerando os estados definidos para os cliente, tanto Web quanto o Raspberry Pi, e a natureza *stateless* do REST, fez-se necessário grande atenção ao sincronismo das mensagens e o desenvolvimento de mecanismos para checagem do estado atual da aplicação. O fluxo rotineiro de mensagens do sistema para a gravação de áudio pode ser observado na Figura 14.

A interação com o sistema começa a partir da autenticação do professor. Após a listagem e seleção do curso desejado passa-se para a solicitação de início da sessão de gravação.

A partir deste ponto, o cliente web fica aguardando o início efetivo da gravação pelo Raspberry, checando periodicamente a API por esta confirmação. Do mesmo modo, o cliente de gravação checa constantemente o servidor por alguma solicitação de início e, após receber a confirmação, inicia de fato a gravação do áudio e envia à API uma mensagem efetivando o início da gravação.

Com a gravação sendo realizada, o Raspberry Pi precisa continuar constantemente checando a API por alguma solicitação de parada. Quando a solicitação para parar a gravação é enviada, o cliente web passa a aguardar pela confirmação do término e, o cliente Raspberry Pi, recupera o áudio gravado e o envia para o servidor, encerrando assim a sessão de gravação. Deste ponto em diante, o áudio gravado fica disponível para reprodução no cliente Web, juntamente com os comentários eventualmente adicionados pelos alunos à sessão de gravação.

### 5.2.5 Clientes Desenvolvidos

O sistema para gravação de áudios de sala de aula é composto basicamente por três componentes, a API, que fornece os serviços básicos e específicos, o cliente Web, que fornece interface para o usuário e o dispositivo de Internet das coisas, o cliente Raspberry Pi, que gerencia o mecanismo de gravação em si. A baixo tem-se detalhado os detalhes de implementação dos dois clientes.

Figura 15 – Interface Web em Smartphone

The interface consists of several components:

- Header:** "Teacher: thiago", "User: thiago", and a "Logout" button.
- Recording Section:** "Recording", "User: thiago", "Logout" button, "Status: Idle", "Start" button, "Stop" button, and a recording card with fields for "Id" (58588fabc9d), "Date" (19/12/2016 2), "Audio" (play icon), and "Status" (ended).
- Add Note Section:** "Add Note", "Text" input field, and "Add" button.
- Recordings List Section:** "Recordings List" and a table with columns "Date/Time", "User", and "Note".

Date/Time	User	Note
19/12/2016 22:57:42	thiago	teste
14/01/2017 16:16:45	thiago	Seconde not
14/01/2017 16:17:28	thiago	Other annota

Fonte: Elaborado pelo autor

### 5.2.5.1 Cliente Web

O cliente Web é a porta de entrada do usuário no sistema, é ele que provê a interface para professores e alunos se autenticarem, buscarem seus cursos, adicionarem anotações, iniciarem sessões de gravação e etc. É este módulo que transforma os dados e ações da API em interfaces amigáveis ao ser humano, que se adaptam a diferentes tipos de tela, possibilitando a utilização do sistema tanto em computadores como em *smatphones* e tablets. A Figura 15 traz algumas telas do sistema exibidas em um dispositivo móvel.

O desenvolvimento deste módulo deu-se utilizando exclusivamente as linguagens HTML, CSS e JavaScript. Para as requisições à API, utiliza-se as funções AJAX da biblioteca JQuery JS e, para uma interface amigável e responsiva, faz-se o uso do Bootstrap CSS.

### 5.2.5.2 Cliente de Gravação

O cliente gravação é composto por *hardware* e *software*. Utiliza-se um Raspberry Pi, um computador pequeno e barato, construído em uma única placa e desenvolvido com fins educacionais (RICHARDSON; WALLACE, 2012). Para compor a solução, adicionam-se os seguintes periféricos: um receptor Wi-Fi USB, um microfone USB, um *hub* USB e uma fonte de alimentação. A Figura 16 mostra este dispositivo e seus periféricos.

No que diz respeito ao software, o mesmo foi desenvolvido através de *scripts* da

Figura 16 – Dispositivo e seus periféricos



Fonte: Elaborado pelo autor

linguagem PHP, executados via linha de comando. Configurou-se, no Linux do Raspberry, através do agendador de tarefas Cron, para que os programa seja executado sempre que o dispositivo é iniciado. Desse forma, após o carregamento do sistema operacional, o dispositivo de gravação já começa a esperar, e buscar na API, por solicitações de início da gravação.

### **5.2.6 Considerações finais sobre o Cliente de Gravação**

Neste capítulo apresentou-se o cliente para gravação de áudios de aula. Tal cliente foi desenvolvido em cima de uma arquitetura base para aplicações de Internet das Coisas, detalhada no Capítulo 5.1. Novas funcionalidades específicas do domínio de negócio desta aplicação foram desenvolvidas e adicionadas às já fornecidas pela API original, seguindo o padrão arquitetural proposto.

Acredita-se que, a gravação e disponibilização de áudios de aula, juntamente com os comentários adicionados pelos alunos, venha a ser uma importante ferramenta para a elucidação de dúvidas e para revisão de conteúdos por parte dos estudantes. Espera-se que este seja o passo inicial para o desenvolvimento de uma sala de aula ubíqua nos moldes da arquitetura proposta.



## 6 RESULTADOS

Para a definição da amostra foram selecionados alunos das turmas de Programação Orientada a Objetos e Programação de Dispositivos Móveis do primeiro semestre de 2017, totalizando 43 (quarenta e três) estudante. A preparação do ambiente do experimento se deu através da implantação do protótipo do software em um servidor web e da configuração da aplicação e do cliente Raspberry Pi, com o cadastramento dos usuários e turmas.

Para a validação da solução, foi ministrada aos alunos, em duas turmas, uma breve aula sobre o tema "Internet das Coisas". Após treinamento básico da ferramenta, os alunos foram instruídos a utilizar o software durante, e após a aula, para adicionarem anotações, comentários e revisarem o assunto, através da reprodução da gravação no cliente web. Vencida esta etapa, foi enviado por e-mail aos alunos questionário do Google Forms, presente no Apêndice A, e solicitado que o mesmo fosse respondido de acordo com a experiência que cada um teve ao utilizar o sistema. Os resultados obtidos para cada pergunta é apresentado abaixo.

### 6.1 RESULTADOS DO QUESTIONÁRIO

Iniciou-se o questionário indagando ao aluno sobre a usabilidade do sistema, qual a facilidade em acessar e utilizar o software. Obteve-se um resultado expressamente positivo, com mais de 72% de usuários considerando o sistema muito fácil de ser utilizado, como pode ser observado na Figura 17.

Figura 17 – Respostas da primeira pergunta

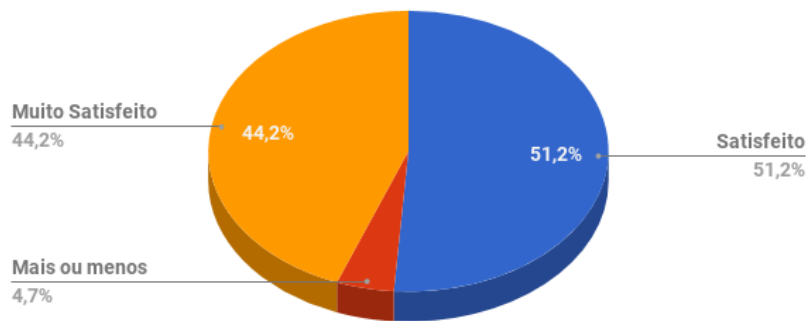


Fonte: Elaborado pelo autor

Seguindo, os alunos foram questionados quanto a responsividade da aplicação, se a interface gráfica estava ajustada ao tamanho da tela utilizada para acessar ao sistema. Conforme mostra a Figura 18, identificou-se que mais de 95% dos entrevistados se disseram satisfeitos, ou muito satisfeitos, mostrando que a aplicação se comportou de maneira adequada.

Figura 18 – Respostas da segunda pergunta

A interface gráfica da aplicação estava compatível com o tamanho da tela do seu dispositivo?

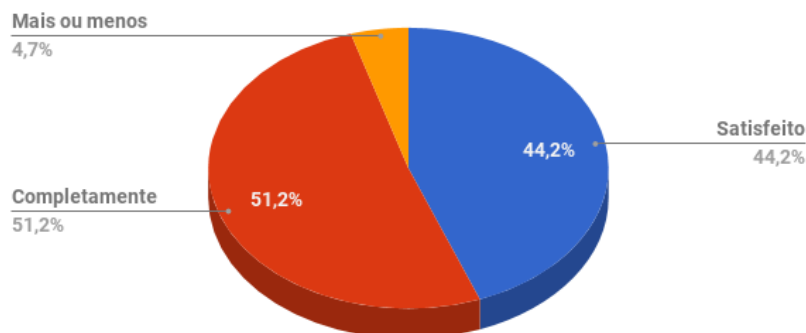


Fonte: Elaborado pelo autor

Na terceira pergunta, avaliou-se o êxito da funcionalidade de adição de anotações durante a sessão de gravação, durante a aula. Novamente, como pode ser observado na Figura 19, mais de 95% dos usuário se sentiram satisfeitos ou completamente satisfeitos.

Figura 19 – Respostas da terceira pergunta

Como foi a sua experiência ao adicionar anotações durante a sessão de gravação da aula?



Fonte: Elaborado pelo autor

Em seguida, os alunos foram questionados quanto a qualidade do áudio gravado. Neste quesito, obteve-se mais de 90% de respostas "muito bom" ou "bom", mostrando que, mesmo com o limitações de hardware, foi possível captar um som de boa qualidade. Este resultado por ser observado na Figura 20.

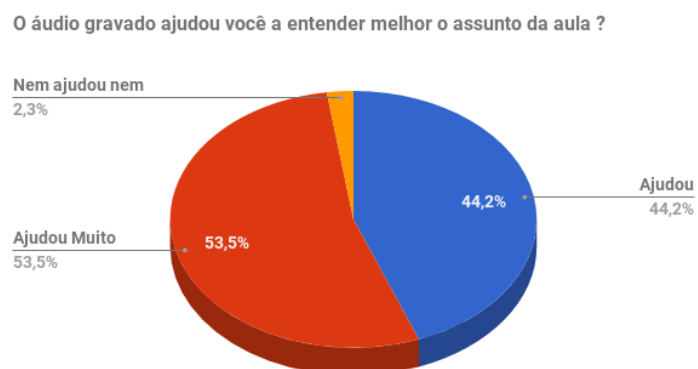
Figura 20 – Respostas da quarta pergunta



Fonte: Elaborado pelo autor

A partir da quinta pergunta, tem-se uma avaliação do quanto a utilização do sistema efetivamente auxiliou o aluno no processo de aprendizagem. Aqui, é perguntado se o áudio gravado ajudou o aluno a entender melhor o assunto da aula ministrada, e, conforme a Figura 21, observa-se que 97,7% consideraram que as gravação os ajudaram, ou ajudaram muito, a compreender o assunto abordado na aula.

Figura 21 – Respostas da quinta pergunta



Fonte: Elaborado pelo autor

Nesta pergunta, é avaliado o quanto as anotações feitas pelo próprio aluno, durante a gravação da aula, o ajudaram e entender melhor o assunto ministrado. Novamente, mais de 90% responderam que as anotações ajudaram, ou ajudaram muito, como pode ser observado na Figura 22.

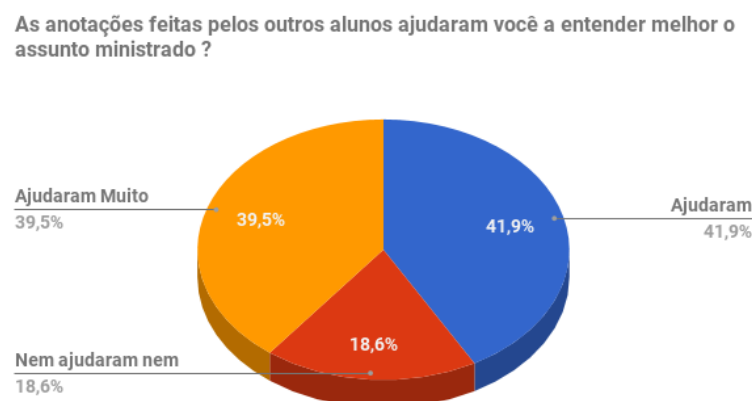
Figura 22 – Respostas da sexta pergunta



Fonte: Elaborado pelo autor

Na sétima pergunta, tentou-se identificar o quanto as anotações de outros alunos influenciaram no processo de aprendizagem do entrevistado. Neste quesito, cerca de 18% se disseram indiferentes quanto às anotações de outros alunos, enquanto cerca de 81% consideraram que estas anotações os ajudaram ou ajudaram muito, como é mostrado na Figura 23.

Figura 23 – Respostas da sétima pergunta



Fonte: Elaborado pelo autor

Por fim, foi apresentada aos alunos uma questão discursiva, de campo aberto, onde os mesmos poderiam deixar críticas e sugestões ao sistema apresentado. De maneira geral, os entrevistados registraram congratulações e sugestões, que serão acatadas como trabalhos futuros, como por exemplo, adicionar gravação de imagens (vídeo), capturar a tela do computador do professor (slides), oferecer a opção de download da mídia gravada e desenvolver um aplicativo para dispositivos móveis.

## 6.2 CONSIDERAÇÕES SOBRE OS RESULTADOS

Após o desenvolvimento e implantação do protótipo, o mesmo foi apresentado a 43 alunos da Universidade Estadual do Maranhão, regularmente matriculados no curso de Engenharia da Computação. Após breve treinamento, os mesmos fizeram uso do sistema durante uma aula sobre Internet das Coisas, adicionando anotações e revisando o conteúdo ministrado através do áudio gravado. Findada esta etapa, os usuários responderam a um questionário elaborado com sete perguntas objetivas, cujos resultados sumarizados encontram-se na Tabela 2.

Tabela 2 – Resultado dos questionários aplicados

Questão	Respostas				
	Muito Satisfeito	Satisfeito	Regular	Insatisfeito	Muito Insatisfeito
#					
1	72,1%	23,3%	4,7%	0,0%	0,0%
2	44,2%	51,2%	4,7%	0,0%	0,0%
3	51,2%	44,2%	4,7%	0,0%	0,0%
4	44,2%	48,8%	7,0%	0,0%	0,0%
#	<b>Ajudou Muito</b>	<b>Ajudou</b>	<b>Neutro</b>	<b>Atrapalhou</b>	<b>Atrapalhou Muito</b>
5	53,5%	44,2%	2,3%	0,0%	0,0%
6	41,9%	53,5%	4,7%	0,0%	0,0%
7	39,5%	41,9%	18,6%	0,0%	0,0%

Analisando os resultados obtidos, identifica-se, por parte dos entrevistados, uma boa aceitação do sistema proposto. Observa-se que, tanto nas questões mais direcionadas às características técnicas do sistema, quanto as que dizem respeito ao entendimento em si da aula ministrada, os números são notoriamente satisfatórios.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Esta pesquisa teve como objetivo propor um sistema computacional que utilize dispositivos de Internet das Coisas como ferramenta de suporte a professores e alunos com a gravação de áudios de aulas. Neste sentido, esperava-se que, com os áudios gravados e as anotações realizadas, os alunos tivessem mais facilidade para aprenderem os assuntos ministrados em sala de aula.

Para atingir o objetivo geral, a pesquisa atendeu todos os objetivos específicos que foram propostos inicialmente para o desenvolvimento do sistema proposto, utilizando uma plataforma pensada para atender a todos os desafios da Internet das Coisas. Neste contexto, propôs-se uma arquitetura orientada a serviços, que garante a alta disponibilidade, tolerância a falhas e escalabilidade, bem como um cliente web e Raspberry Pi, para a efetiva gravação dos áudios e interação com os usuários.

Para avaliar a contribuição do sistema, o mesmo foi disponibilizado a alunos do curso de Engenharia da Computação durante uma aula sobre o tema "Internet das Coisas". Neste momento, os estudantes utilizaram as funcionalidades desenvolvidas, adicionando e visualizando comentários e reproduzindo o áudio gravado da aula e, em momento subsequente, foram apresentados a um questionário de avaliação da solução.

As perguntas apresentadas no questionário podem ser divididas em duas categorias, as relativas aos aspectos técnicos da solução e as direcionadas à percepção de aprendizagem. No primeiro caso, como foi observado nas Figura 17, Figura 18, Figura 19 e Figura 20, os entrevistados, em sua ampla maioria, se sentiram satisfeitos, ou muito satisfeito, com o sistema desenvolvido, mostrando que o mesmo atendeu bem aos requisitos que se foram avaliados, tais quais a usabilidade, responsividade, registro de anotações e gravação e reprodução de áudios.

No que diz respeito à percepção de aprendizagem, da quinta à sétima questão, os alunos foram questionados sobre o quanto o sistema os ajudou a entenderem melhor o assunto ministrado na aula. Neste caso, novamente, a maioria dos estudantes respondeu de forma positiva, indicando que o sistema ajudou, ou ajudou muito, no entendimento do conteúdo, conforme pôde ser observado nas Figura 21, Figura 22 e Figura 23. A Tabela 2 trouxe ainda um panorama geral de todas as respostas objetivas aplicadas.

Ao final do formulário, foi disponibilizado aos alunos campo de texto livre para expressarem sugestões e críticas à pesquisa e ao sistema desenvolvido. Observou-se que, além de congratulações, foram deixadas diversas sugestões de melhorias e novas funcionalidades de forte

relevância para o prosseguimento deste trabalho, entre elas o desenvolvimento de um aplicativo para dispositivos móveis e a captura da tela do professor durante a aula.

Adicionalmente às sugestões deixadas pelos entrevistados, vislumbra-se como trabalhos futuros a inclusão no mínimo das seguintes funcionalidades ao sistema:

- Utilização de padrões de anotações;
- Anotações via mensagem de voz;
- Sincronização entre as anotações e o áudio gravado;
- Registro de anotações *offline*;
- Captura de vídeo da aula;
- Transmissão ao vivo.

Portanto, após análises e discussões conclui-se que, considerando o espaço amostral utilizado, o sistema proposto para gravação de áudio de aula por meio de dispositivos de internet das coisas, atendeu aos objetivos elencados, no sentido de ser uma ferramenta de suporte ao aprendizado dos alunos, facilitando o entendimento dos mesmos aos assuntos ministrados. Desse modo, acredita-se que o sistema se apresenta como uma relevante contribuição para a área de Informática na Educação e em muitos outros contextos. A pesquisa desenvolveu um sistema que fornece ao aluno novos meios para elucidação de dúvidas e revisão de conteúdos.

## REFERÊNCIAS

- ARAUJO, R. B. de. Computação ubíqua: Princípios, tecnologias e desafios. In: **XXI Simpósio Brasileiro de Redes de Computadores**. [S.l.: s.n.], 2003. v. 8, p. 11–13.
- ATZORI, L.; IERA, A.; MORABITO, G. The internet of things: A survey. **Computer networks**, Elsevier, v. 54, n. 15, p. 2787–2805, 2010.
- BAHREE, A.; MULDER, D.; CICORIA, S.; PEIRIS, C.; PATHAK, N. **Pro WCF: practical Microsoft SOA implementation**. [S.l.]: Apress, 2007.
- CHODOROW, K. **MongoDB: the definitive guide**. [S.l.]: "O'Reilly Media, Inc.", 2013.
- CROCKFORD, D. The application/json media type for javascript object notation (json). 2006.
- FIELDING, R. T. **Architectural styles and the design of network-based software architectures**. Tese (Doutorado) — University of California, Irvine, 2000.
- GARTNER INC. Hype cycle for emerging technologies 2016. 2016.
- GIUSTO, D.; IERA, A.; MORABITO, G.; ATZORI, L. **The internet of things: 20th Tyrrhenian workshop on digital communications**. [S.l.]: Springer Science & Business Media, 2010.
- 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.
- GUINARD, D.; TRIFA, V. Towards the web of things: Web mashups for embedded devices. In: **Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences)**. Madrid, Spain: [s.n.], 2009.
- GUINARD, D.; TRIFA, V.; KARNOUSKOS, S.; SPIESS, P.; SAVIO, D. Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. **IEEE transactions on Services Computing**, IEEE, v. 3, n. 3, p. 223–235, 2010.
- GUINARD, D.; TRIFA, V.; WILDE, E. A resource oriented architecture for the web of things. In: IEEE. **Internet of Things (IOT), 2010**. [S.l.], 2010. p. 1–8.
- HAN, J.; HAIHONG, E.; LE, G.; DU, J. Survey on nosql database. In: IEEE. **Pervasive computing and applications (ICPCA), 2011 6th international conference on**. [S.l.], 2011. p. 363–366.
- HEO, Y. J.; OH, S. M.; CHIN, W. S.; JANG, J. W. A lightweight platform implementation for internet of things. In: IEEE. **Future Internet of Things and Cloud (FiCloud), 2015 3rd International Conference on**. [S.l.], 2015. p. 526–531.
- KOVATSCH, M.; MAYER, S.; OSTERMAIER, B. Moving application logic from the firmware to the cloud: Towards the thin server architecture for the internet of things. In: IEEE. **Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on**. [S.l.], 2012. p. 751–756.



- LIU, Y.; LIU, X.; MA, Y.; LIU, Y.; ZHENG, Z.; HUANG, G.; BLAKE, M. B. Characterizing restful web services usage on smartphones: A tale of native apps and web apps. In: IEEE. **Web Services (ICWS), 2015 IEEE International Conference on**. [S.l.], 2015. p. 337–344.
- 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.
- NAYAK, A.; PORIYA, A.; POOJARY, D. Type of nosql databases and its comparison with relational databases. **International Journal of Applied Information Systems**, v. 5, n. 4, p. 16–19, 2013.
- PINTUS, A.; CARBONI, D.; PIRAS, A. Paraimpu: a platform for a social web of things. In: ACM. **Proceedings of the 21st International Conference on World Wide Web**. [S.l.], 2012. p. 401–404.
- RICHARDSON, L.; RUBY, S. **RESTful web services**. [S.l.]: "O'Reilly Media, Inc.", 2008.
- RICHARDSON, M.; WALLACE, S. **Getting started with raspberry PI**. [S.l.]: "O'Reilly Media, Inc.", 2012.
- SAMPIERI, R. H.; COLLADO, C. F.; LUCIO, P. B. Metodología de la investigación. **La Habana: Editorial Félix Varela**, v. 2, 2003.
- SANTAELLA, L. Desafios da ubiquidade para a educação. **Revista Ensino Superior Unicamp**, v. 9, p. 19–28, 2013.
- SATYANARAYANAN, M. Pervasive computing: Vision and challenges. **IEEE Personal communications**, IEEE, v. 8, n. 4, p. 10–17, 2001.
- SAUDATE, A. **REST: Construa API's inteligentes de maneira simples**. [S.l.]: Editora Casa do Código, 2014.
- SOMMERVILLE, I. **Engenharia de Software**. 9. ed. [S.l.]: Pearson, 2011.
- SPIESS, P.; KARNOUSKOS, S.; GUINARD, D.; SAVIO, D.; BAECKER, O.; SOUZA, L. M. S. D.; TRIFA, V. Soa-based integration of the internet of things in enterprise services. In: IEEE. **Web Services, 2009. ICWS 2009. IEEE International Conference on**. [S.l.], 2009. p. 968–975.
- W3C, W. W. W. C. **Simple Object Access Protocol (SOAP) 1.1**. 2000.
- WAGH, K.; THOOL, R. A comparative study of soap vs rest web services provisioning techniques for mobile host. **Journal of Information Engineering and Applications**, v. 2, n. 5, p. 12–16, 2012.
- WEISER, M. Some computer science issues in ubiquitous computing. **Communications of the ACM**, ACM, v. 36, n. 7, p. 75–84, 1993.
- XUE, R.; WANG, L.; CHEN, J. Using the iot to construct ubiquitous learning environment. In: IEEE. **Mechanic Automation and Control Engineering (MACE), 2011 Second International Conference on**. [S.l.], 2011. p. 7878–7880.
- ZIMMERMANN, O.; TOMLINSON, M.; PEUSER, S. **Perspectives on Web Services: Applying SOAP, WSDL and UDDI to Real-World Projects**. [S.l.]: Springer Science & Business Media, 2012.

## **APÊNDICES**

## APÊNDICE A – Formulário Aplicado aos Alunos

# Avaliação Alunos

Formulário de Avaliação do Sistema E-Things, para gravação de áudios de aula

\*Obrigatório

Foi fácil acessar e utilizar o sistema disponibilizado? \*

- Muito Fácil
- Fácil
- Nem fácil nem difícil
- Difícil
- Muito Difícil

A interface gráfica da aplicação estava compatível com o tamanho da tela do seu dispositivo? \*

- Muito Satisfeito
- Satisfeito
- Mais ou menos satisfeito
- Insatisfeito
- Muito Insatisfeito

Como foi a sua experiência ao adicionar anotações durante a sessão de gravação da aula? \*

- Completamente Satisfeito
- Satisfeito
- Mais ou menos satisfeito
- Insatisfeito
- Completamente Insatisfeito

Quanto a qualidade do áudio gravado, qual a sua opinião? \*

- Muito Bom
- Bom
- Regular
- Ruim
- Muito Ruim

O áudio gravado ajudou você a entender melhor o assunto da aula? \*

- Ajudou Muito
- Ajudou
- Nem ajudou nem atrapalhou
- Atrapalhou
- Atrapalhou Muito

As suas anotações, feitas durante a aula, ajudaram você a entender melhor o assunto ministrado ? \*

- Ajudaram Muito
- Ajudaram
- Nem ajudaram nem atrapalharam
- Atrapalharam
- Atrapalharam muito

As anotações feitas pelos outros alunos ajudaram você a entender melhor o assunto ministrado ? \*

- Ajudaram Muito
- Ajudaram
- Nem ajudaram nem atrapalharam
- Atrapalharam
- Atrapalharam muito

Faça crítica e sugestões ao sistema apresentado

Sua resposta

ENVIAR

Nunca envie senhas pelo Formulários Google.

Este conteúdo não foi criado nem aprovado pelo Google. Denunciar abuso - Termos de Serviço - Termos Adicionais

Google Formulários

## APÊNDICE B – Definição da API

Verbo	End-Point	Payload (JSON)	Retorno	Descrição
POST	/auth	username: "username", password: "123"	200 OK id: 1 name: "Fulano de Tal", email: "fulano@mail.com", token: "stredfygblçk986y&oijng9e87%\$#ur3345"	Realiza o login do usuário no sistema e retorna um token de autenticação válido
			401 Não autorizado	
DELETE	/auth		200 OK	Realiza o logout do usuário do sistema. Inativa todos os seus tokens.
			500 Erro do servidor	
POST	/user	name: "Fulano de Tal" username: "username" e-mail: "fulano@mail.com" password: "pass"	201 Created id: 1 name: "Thiago Nunes de Sousa", username: "thiagons"	Cria um novo usuário
			412 Email ou username já utilizado	
PUT	/user	username: "thiagotns" name: "Thiago Nunes", email: "thiago@email.com"	200 OK	Altera as informações de um usuário.
			412 Email já utilizado por outro usuário	
PUT	/user/password	oldPassword: 123, newPassword: 321	200 OK	Altera a senha do usuário
			412 Senha antiga não confere com a atual	
GET	/user/{username}		200 OK Json com dados do usuário	Recupera as informações do usuário com username informado
			404 Usuário não encontrado	
DELETE	/user	password: 123	200 OK	Deleta a conta do usuário logado. É preciso confirmar a senha.
			412 Senha incorreta	
POST	/user/{username}/follow		200 OK	Usuário logado começa a seguir o usuário de username passado
			404 Usuário para seguir não encontrado	

DELETE	/user/{username}/follow		200 OK	Parar de seguir o usuário de username passado
			404 Registro não encontrado	
POST	/course	name: "Test Course"	200 OK	Cria um novo curso e associa o usuário criador como professor
			500 Erro ao processar a operação	
			200 OK	
PUT	/course/id	name: "New Name"	403 Não permitido	Altera as informações do curso Permitido somente para os professores
			200 OK	
			200 OK	
GET	/course		200 OK	Busca por todos os cursos
			Json com as informações de todos os cursos	
			404 Nenhum curso encontrado	
GET	/course/{id}		200 OK	Buscar pelo curso de ID passado
			Json com as informações do curso	
			404 Nenhum curso encontrado	
GET	/course/{id}/teacher		200 OK	Retorna a lista de professores do curso de id passado
			Json com lista das informações dos professores	
			500 Erro ao processar	
POST	/course/{id}/teacher/{user_id}		201 Created	Vincula o usuário de user_id passado como professor de um curso
			412 Pré-condição não atendida	
			200 OK	
GET	/course/{id}/enrollment		200 OK	Retorna as informações dos usuários matriculados no curso
			Lista com informações dos usuários matriculados	
			404 Não encontrado	
POST	/course/{id}/enrollment		201 Matriculado	Matricula o usuário logado a um curso
			404 Curso não encontrado	
			412 Pré-condição não atendida	
DELETE	/course/{id}/enrollment		200 OK	Cancela a matrícula do usuário logado em um curso
			500 Erro ao processar a operação	

DELETE	/course/{id}		200 OK 412 Pré-condição não atendida	Exclui um curso. Só é permitido para o admin do curso.
DELETE	/course/{id}/teacher/{user_id}		200 OK 412 Pré-condição não atendida	Remove o vínculo de professor do usuário de id_id_user do curso
POST	/course/{id}/resource/{id_resource}		201 Criado 412 Pré-condição não atendida	Vincula o recurso id_resource ao curso
DELETE	/course/{id}/resource/{id_resource}		200 OK 412 Pré-condição não atendida	Desvincula o recurso id_resource do curso
POST	/resource	JSON com as informações do recurso	201 Criado 500 Erro do servidor	Cria um novo recurso.
PUT	/resource/{id}	JSON com as informações para atualizar	201 Criado 500 Erro do servidor 404 Recurso não encontrado	Atualiza um recurso
GET	/resource?key=value		200 OK Json dos recursos encontrados 404 Não encontrado	Busca por um recurso usando como filtro a chave e valor passados como parâmetro
GET	/resource/{id}		200 OK Json com as informações do recurso 404 Não encontrado	Recupera o recurso de id informado
DELETE	/resource/{id}		200 OK 404 Não encontrado	Exclui um recurso do sistema



## APÊNDICE C – Código-Fonte dos métodos para iniciar gravação e realizar envio da mídia no Raspberry Pi

```
1 function start_recording($id_rec){
2
3     if (!file_exists('recs') ) {
4         mkdir('recs', 0777, true);
5     }
6
7     $command = "arecord -D sysdefault:CARD=TP6920 -r 48000 -f
8         S16_LE -t raw | lame -r -s 48 --signed --little-
9         endian -m m --vbr-new -V 0 - recs/" . $id_rec . ".mp3"
10        ;
11
12    exec('nohup '. $command .' > /dev/null 2>&1 & echo $!');
13    return true;
14 }
15
16 function upload($id_course, $id_rec, $file_path, $url,
17     $token){
18
19     echo date("Y-m-d H:i:s") . " Carregando o arquivo...\n"
20        ;
21
22     $file = file_get_contents($file_path);
23     $base64 = base64_encode($file);
24
25     $data = [
26         'content' => $base64
27     ];
28
29     echo date("Y-m-d H:i:s") . " Iniciando o envio...\n";
30 }
```

```
26 $curl = curl_init();
27
28 curl_setopt($curl, CURLOPT_URL, $url . "/courses/" .
    $id_course . "/recording/" . $id_rec . "/upload");
29 curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
30 curl_setopt($curl, CURLOPT_CUSTOMREQUEST, 'PUT');
31 curl_setopt($curl, CURLOPT_POSTFIELDS, json_encode($data)
    );
32 curl_setopt($curl, CURLOPT_HTTPHEADER, array('
    Authorization:' . $token, 'Content-type: application/
    json'));
33
34 $server_output = curl_exec ($curl);
35
36 if ($server_output === false) {
37     $server_output = curl_getinfo($curl);
38     echo "\n\nCurl Error: " . serialize($server_output);
39     return NULL;
40 }
41
42 $httpcode = curl_getinfo($curl, CURLINFO_HTTP_CODE);
43
44 if($httpcode == 404){
45     return NULL;
46 }
47
48 echo date("Y-m-d H:i:s") . " Envio Finalizado!\n";
49
50 $obj = json_decode($server_output);
51
52 return $obj;
53 }
```

APÊNDICE D – Código-Fonte dos *end-points* da API para envio e recuperação de gravações

```
1 router.put('/:id_course/recording/:id_rec/upload', function(req , res
  , next){
2
3   Course.findOne({_id: req.params.id_course}, function(err, course){
4     if(err){
5       res.json(err);
6       return;
7     }
8
9     if(course == null){
10      res.status(404).end("Course Not Found");
11      return;
12    }
13
14    Recording.findOne({_id: req.params.id_rec, course: course},
      function(err, rec){
15      if(err){
16        res.json(err);
17        return;
18      }
19
20      if(rec == null){
21        res.status(404).end("Recording Not Found");
22        return;
23      }
24
25      if(rec.status != 'stoped'){
26        res.status(412).end('Not Stopped!');
27        return;
28      }
29
30      var buffer = new Buffer(req.body.content, "base64");
31      rec.content = buffer;
32      rec.status = 'ended';
33
34      rec.save(function(err, rec_resp){
35        if (err)
36          return res.end(JSON.stringify(err));
```

```
37
38     var obj = JSON.parse(JSON.stringify(rec_resp));
39     delete obj.content;
40
41     res.status(200).json(obj);
42   });
43 });
44 });
45 });
46
47 router.get('/:id_course/recordings', function(req , res, next){
48
49   Course.findOne({_id: req.params.id_course}, function(err, course){
50     if(err){
51       res.json(err);
52       return;
53     }
54
55     if(course == null){
56       res.status(404).end();
57       return;
58     }
59
60     with_content = req.query.content == null ? false : req.query.
61       content;
62
63     Recording.find({course: course}, 'status created_at', function(
64       err, list){
65       if(err){
66         res.json(err);
67         return;
68       }
69
70       if(list == null){
71         res.status(404).end();
72         return;
73       }
74
75       var retorno = [];
```

```
74
75     for (var i = 0; i < list.length; i++) {
76         //faz copia do objeto para poder alterar
77         var obj = JSON.parse(JSON.stringify(list[i]));
78
79         if(with_content){
80             temp = new Buffer(list[i].content, "base64");
81             obj.content = temp.toString();
82         } else {
83             delete obj.content;
84         }
85
86         retorno.unshift(obj);
87     }
88
89     res.json(retorno);
90 });
91 });
92 });
```