INSTITUTO FEDERAL DE SANTA CATARINA

LETICIA APARECIDA COELHO

Infraestrutura de Internet das Coisas aplicada a laboratórios remotos usando WebRTC e microsserviços

São José - SC

novembro/2018

INFRAESTRUTURA DE INTERNET DAS COISAS APLICADA A LABORATÓRIOS REMOTOS USANDO WEBRTC E MICROSSERVIÇOS

Pré-projeto do trabalho de conclusão de curso apresentado à Coordenação de Engenharia de Telecomunicações do Instituto Federal de Santa Catarina para a obtenção do diploma Bacharel em Engenharia de Telecomunicações.

Orientador: Ederson Torresini

Coorientador: Joaquim Manoel Gonçalves

São José - SC

novembro/2018

RESUMO

Os laboratórios remotos surgiram com o propósito de disponibilizar os recursos utilizados para o aprendizado de disciplinas e experimentos técnicos de forma remota, facilitando a aprendizagem e incentivando a visão crítica para estudantes da área técnica. A aplicação é especificada com base em conceitos de *Internet of Things* (IoT), através de uma arquitetura distribuída, com microsserviços, possibilita escalabilidade para o acesso de usuários, e a utilização de WebRTC para transmissão da imagem e vídeo em tempo real, possibilita a visualização do experimento sem a necessidade de instalação de aplicativos extras. Assim, o resultado é uma infraestrutura com especificação genérica dos dados, que possibilita o gerenciamento de laboratórios remotos de diversas áreas de estudo, e disponibiliza a utilização do experimento de forma fácil e fluida.

Palavras-chave: Internet das coisas (Iot). WebRTC. Serviços de mensagem.

LISTA DE ILUSTRAÇÕES

Figura 1 – Soluções baseadas em IoT. *Fonte: Próprio autor, baseado em (HöLLE, 2014) e (DESAI, 2015).* 17
Figura 2 – Arquitetura IoT. *Fonte: Próprio autor, baseado em (ITU, 2012), (DESAI, 2015),(BILAL, 2017).* 18
Figura 3 – Arquitetura IoT - Comunicação. *Fonte: Próprio autor, baseado em (IBM,).* 19
Figura 4 – Arquitetura IoT - Visão específica. *Fonte: Próprio autor, baseado em (ITU, 2012).* . . . 20
Figura 5 – Exemplo *Uniform Resource Identifier* (URI). *Fonte: Berners-Lee, Fielding e Masinter (2005).* 23
Figura 6 – Arquitetura *WebRTC Fonte: Grigorike (2013).* 25
Figura 7 – Estrutura de laboratório remoto. *Fonte: Próprio autor baseado em VISIR (2018).* . . . 26
Figura 8 – Cenário 1 - Arquitetura proposta. *Fonte: Próprio autor.* 29
Figura 9 – Cenário 2 - Arquitetura proposta. *Fonte: Próprio autor.* 30

LISTA DE TABELAS

Tabela 1 – Cronograma das atividades previstas 15

LISTA DE CÓDIGOS

Código 2.1 – Botão em HTML 22
Código 2.2 – Estilo para o botão em CSS 22
Código 2.3 – Websocket - Servidor 24
Código 2.4 – Websocket - Cliente 24

LISTA DE ABREVIATURAS E SIGLAS

IFSC *Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina* 13

IoT *Internet of Things* 2

HTML *Hypertext Markup Language* 22

CSS *Cascading Style Sheets* 22

URI *Uniform Resource Identifier* 3

URL *Uniform Resource Locator* 23

HTTP *Hypertext Transfer Protocol* 23

API *Application Programming Interface* 15

TCP *Transmission Control Protocol* 24

IP *Internet Protocol* 24

ITU *International Telecommunication Union* 17

M2M *Machine-to-Machine* 17

WSN *Wireless sensor network* 19

MTC *Machine-type-Communication* 18

WAN *Wide Area Network* 18

PAN *Personal Area Network* 18

LAN *Local Area Network* 18

Wi-Fi *Wireless Fidelity* 18

UWB *Ultra WideBand* 18

6LoWPAM *IPv6 over Low power Wireless Personal Area Networks* 18

LTE *Long Term Evolution* 18

3G *Third-Generation Cell-Phone Technology* 18

UI *User Interface* 19

W3 *World Wide Web* 22

URN *Uniform Resource Name* 22

BCP35 *Best Current Practice* 23

HTTPbis *HTTP Group Work* 24

SGML *Standard Generalized Markup Language* 22

JSON *JavaScript Object Notation* 24

RTC *Real Time Communications* 25

NAT *Network Address Translation* 25

REST *Representational State Transfer* 29

SUMÁRIO

1 INTRODUÇÃO . . . 13
1.1 Objetivos . . . 13
1.2 Justificativa . . . 14
1.3 Metodologia . . . 15

2 FUNDAMENTAÇÃO TEÓRICA . . . 17
2.1 Internet das Coisas . . . 17
2.1.1 Arquitetura em camadas . . . 18
2.1.2 Arquitetura em serviços . . . 19
2.1.3 Arquitetura de Software . . . 21
2.2 Arquitetura Web . . . 22
2.2.1 Formatação . . . 22
2.2.2 Endereçamento . . . 22
2.2.3 Transferência de dados . . . 23
2.2.4 *WebSocket* . . . 24
2.2.5 WebRTC . . . 25
2.3 Laboratórios remotos . . . 26

3 PROPOSTA . . . 29
3.1 Cenário 1 - Medição de dados . . . 29
3.2 Cenário 2 - Atuação no sistema . . . 30

REFERÊNCIAS . . . 31

1 INTRODUÇÃO

O desenvolvimento de tecnologias que auxiliem no processo educacional sofreu constantes mudanças ao longo da história, provenientes dos avanços tecnológicos e alta utilização da Internet. Diversas ferramentas foram criadas, auxiliando os educadores a disponibilizar os conteúdos teóricos de forma mais criativa e assertiva, objetivando melhor aprendizagem (DINIZ, 2001).

Uma importante motivação para o surgimento de experimentos remotos é o fato de que a carga horária e disposição de equipamentos em laboratórios e bancadas reais tendem a não ser suficientes para o aprendizado dos estudantes durante o período de aula, o que os move, em alguns casos, a realizar apenas estudos teóricos e experimentos em simuladores, reduzindo sua percepção no que diz respeito a respostas reais relativas a teoria aplicada (LIMA, 2017).

O ensino de disciplinas que atuam com grandezas físicas é uma tarefa difícil para as quais são usadas as mais variadas ferramentas, como estudos teóricos, experimentos físicos, simuladores, experimentos remotos e laboratórios remotos. O objetivo principal dessas ferramentas é auxiliar no processo de ensino e aprendizagem dos alunos, relacionado as mais diversas características das teorias aplicadas, em cursos com foco técnico e experimental (LIMA, 2017).

No que se refere a resultado real, ou seja, medições de dados reais provenientes de sensores, um fator importante está relacionado às não-idealidades dos componentes e as condições físicas nas quais o experimento está inserido. Estas características são muito complexas para serem modeladas em um simulador, pois variam conforme o ambiente em que o experimento está instalado. Desse modo, os valores podem apresentar diferenciações devido a fatores temporais (temperatura, umidade) e estruturais (instalação, energia, influência eletromagnética) gerando um aprendizado comprometido nos estudantes devido ao ambiente limitado (BRANCO, 2017).

Infraestruturas de laboratórios remotos possibilitam a visualização de experimentos de forma remota, acionamento e visualização de valores provenientes de experimentos físicos utilizando as premissas de IoT, os quais estão contidos em um dispositivo físico que utiliza protocolos de rede, para a transferência dos dados em tempo real utilizando o protocolos relativos a aplicações Web. Arquitetura deste tipo possibilitam troca de mensagens entre o experimento e o servidor Web, através de serviços de conectividade e servidores de mensagens.

O apoio ao ensino presencial é uma demanda crescente no processo educacional. O *Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina* (IFSC) campus São José possui uma demanda no curso de Refrigeração e Climatização de utilizar soluções remotas que auxiliem os estudantes em períodos fora de classe. O experimento de lampada estroboscópica, tem o propósito de auxiliar no aprendizado de conceitos de motores, possuindo um motor com envólucro transparente e uma lampada estroboscópica.

1.1 Objetivos

Este trabalho apresenta uma visão prática e teórica de infraestrutura de software aplicado a projetos com conceitos de IoT. A partir dos estudos realizados é perceptível que as infraestruturas e cenários de desenvolvimento e disponibilidade de laboratórios remotos possuem diversas limitações técnicas, onde a lógicas de IoT podem auxiliar na solução destes problemas. Este trabalho objetiva propor soluções para solução de problemas inerentes a disponibilidade de arquiteturas para laboratórios remotos.

- Pesquisar uma infraestrutura de software adequada para implementação de experimentos remotos que atuam como apoio ao ensino presencial, propondo uma infraestrutura que utilize abordagens de IoT para solução das dificuldades de desenvolvimento de laboratórios remotos.
- Aplicar a infraestrutura estudada para o experimento de motor com estrutura transparente utilizando lâmpada estroboscópica, com o objetivo de verificar a viabilidade do desenvolvimento de infraestruturas deste tipo aplicadas a experimentos remotos.
- Validar as metodologias desenvolvidas na CTIC do IFSC com o objetivo de incentivar o surgimento de novos experimentos remotos.

1.2 Justificativa

O desenvolvimento de tecnologias que auxiliem no processo de ensino e aprendizagem tem potencial para desenvolver soluções que promovam equidade e qualidade na educação . As macrotêndencias desenvolvidas ao longo da historia para o processo da evolução educacional indicam que diferentes tecnologias foram utilizadas para facilitar o acesso as necessidades dos integrantes do processo educacional, ou seja dos estudantes e professores (PORVIR, 2018).

Atualmente as tendências educacionais qualificam o "*conhecimento nas mãos do usuário*", ou seja, através da Internet os estudantes têm acesso a aplicativos e ferramentas que auxiliam no aprendizado. Tecnologias que não restringem lugar ou hora, além de considerar que diferentes pessoas possuem diferentes ritmos de aprendizado (PORVIR, 2018) e a consideração destes pontos torna o processo educacional mais satisfatório.

A *organização Todos Pela Educação* define como inovação tecnológica projetos que promovam qualidade, ou seja, que expandem o acesso a informação, facilitam o entendimento de conteúdos, despertam a curiosidade do aluno, ampliam as possibilidades de prática de expressão, competências cognitivas e socioemocionais. Considerando como uma das características mais importantes a diferenças no processo de aprendizado individual ((TPE), 2018).

As diferenças individuais no processo de aprendizagem são exploradas através de diversas ferramentas, como jogos, animações, simuladores, videoaulas, (PORVIR, 2018) laboratórios remotos e experimentos remotos. Estas ferramentas possuem a característica principal de necessitarem de infraestruturas de Internet que proporcionem acesso mesmo com baixa conectividade, altas taxas de latência, limitação de instalação de novos programas e limitação de hardware.

A implementação de arquiteturas baseadas em IoT e que funcionem através da Web, possibilitam inovações tecnológicas educacionais, que possuam como característica medições de dados reais, ou seja, laboratórios remotos e experimentos remotos. Isso se deve ao fato de essas infraestruturas possuírem arquiteturas que possibilitam a captação e armazenamento de dados, além de tomada de decisão através da análise dos dados obtidos pela utilização dos projetos (ITU, 2012).

A interconectividade possibilita que as infraestruturas de rede deficitárias não influenciam no funcionamento, e permitem através de acesso remoto considerável reaproveitamento na estrutura física, apoiadas pelas premissas de reutilização e baixa manutenção dos laboratórios remotos e de larga escala e baixo gasto energético das aplicações IoT. Assim sendo, laboratórios remotos implementados com infraestruturas consistentes, como as baseadas em IoT, permitem alto aproveitamento no cenário educacional brasileiro.

Projetos que utilizem desta área de atuação possuem a premissa de realizar melhorias na sociedade, tornando-a mais justa e igualitária (CIEB, 2018). A média de computadores no Brasil é de 7 por escola

(GRACILIANO, 2018), logo o pouco acesso a educação é realidade e necessita de modificações, através da educação para todas as escalas da sociedade, sem considerar posição financeira ou histórico acadêmico, toda uma sociedade pode ser modificada.

1.3 Metodologia

A metodologia do trabalho consiste no desenvolvimento das etapas de estudo, implementação e validação da infraestrutura proposta. A primeira etapa, de estudo, é a análise das melhores ferramentas para implementação de cada uma das partes do projeto, ou seja, caracteriza principalmente o estudo de tecnologias e compatibilidade entre os protocolos/tecnologias definidos. A segunda etapa é constituída no desenvolvimento do projeto, as implementações de uma *Application Programming Interface* (API) REST, uma página Web e um experimento base foram realizadas como prova de conceito para o projeto, a implementação do sistema embarcado e da comunicação entre as partes. A terceira e última etapa conta com a colaboração de empresas parceiras, com a implementação real do experimento, o qual será disponibilizado como serviço pela empresa, possibilitando a validação do funcionamento do experimento.

Tabela 1 – Cronograma das atividades previstas

Etapa	Semanas					
	Fevereiro	Março	Abril	Maio	Junho	Julho
Front-end	√					
Integração	√					
Sistema embarcado		√				
Broker		√				
Integração		√				
WebRTC			√	√		
Integração				√	√	√
Documentação	√	√	√	√	√	√

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Internet das Coisas

A definição de IoT apresentada pela *International Telecommunication Union* (ITU) é de uma infraestrutura global que permite serviços avançados interoperáveis, possuindo a capacidade de conectar o físico ao virtual. Utilizando sistemas com capacidade para interconectividade, que consideram a diversidade de componentes na camada física e monitoram alterações dinâmicas no estado dos dispositivos conectados a rede (ITU, 2012).

Com diferentes características e ambientes de comunicação, as infraestruturas de IoT possuem a premissa de fornecer serviço para aplicações de larga escala (ITU, 2012). Para prover soluções de qualidade é necessário projetar ferramentas e funcionalidades que propiciem uma rede de comunicação entre os dispositivos, independente do aumento de componentes na rede.

Os dispositivos conectados em uma rede IoT utilizam diferentes arquiteturas de intercomunicação, através do qual diversas modularizações podem ser aplicadas. A comunicação pode intercorrer através de um *Gateway*, entre os dispositivos caracterizada pelo conceitos de *Machine-to-Machine* (M2M) e por meio de uma arquitetura modular com ambas as características. Uma infraestrutura ou aplicação IoT possui como premissa elementos físicos - *coisas* - que informem características monitoradas em tempo real para uma rede local ou para a Internet (HöLLE, 2014).

Figura 1 – Soluções baseadas em IoT. *Fonte: Próprio autor, baseado em (HöLLE, 2014) e (DESAI, 2015).*

O conceito de M2M se refere a comunicação somente entre dispositivos que estejam conectados, ou não, em uma rede. Apresenta um contraste com a definição de IoT, porém é definida por aplicações que não possuem intervenção humana denotadas *Machine-type-Communication* (MTC). Abrangendo aplicações da esfera de medição, telemática, monitoramento remoto, segurança e outras que necessitem de captação de dados e tomada de decisão através de sensores e/ou atuadores (HöLLE, 2014) sem ações de usuários.

A utilização de componentes físicos para realização de medições em tempo real é um fator crucial para o crescimento do desenvolvimento de ambientes e aplicações inteligentes, pois para a tomada de decisão é necessário conhecer informações referentes ao entorno e internas a solução (DAS, 2005). Gerando crescente necessidade na miniaturização de componentes para disponibilidade de projetos de uso cotidiano com baixo custo.

2.1.1 Arquitetura em camadas

As diversas aplicações enquadradas ao conceito de IoT possibilitam diferentes modularizações, porém devido as semelhanças é factível definir uma arquitetura em camadas. Admitimos uma arquitetura com camadas semelhantes entre os diversos tipos de aplicações das mais distintas áreas de uso demonstradas na Figura 1. A Figura 2 exibe as camadas principais que englobam o desenvolvimento de arquiteturas IoT, possuindo quatro camadas principais que definem as necessidades principais de aplicações com este conceito.

Figura 2 – Arquitetura IoT. *Fonte: Próprio autor, baseado em (ITU, 2012), (DESAI, 2015),(BILAL, 2017).*

A camada de percepção de dados é a primeira camada, esta ligada a capturar dados obtidos em medições reais de componentes físicos únicos e identificáveis, - por exemplo temperatura, pressão, umidade e entre outros - e transformá-los em informação para transmissão em rede (BILAL, 2017). Possibilita tomada de decisões através de sensores e atuadores, considerando a característica principal de heterogeneidade entre os componentes (ITU, 2012), ou seja, é independente de necessidades técnicas específicas, apenas deve possibilitar a comunicação definida para cada solução.

A camada mais desenvolvida da arquitetura IoT atual é a segunda camada, de rede (GAITAN, 2015), possibilitando a comunicação segura entre a camada de percepção e aplicação (BILAL, 2017). Implementa diferentes topologia de rede como *Wide Area Network* (WAN), *Personal Area Network* (PAN) e *Local Area Network* (LAN), através de tecnologias como *Wireless Fidelity* (Wi-Fi), Ethernet, *Ultra WideBand* (UWB), ZigBee, *IPv6 over Low power Wireless Personal Area Networks* (6LoWPAM), *Long Term Evolution* (LTE), *Third-Generation Cell-Phone Technology* (3G), entre outras padronizadas para o uso de transmissão de dados (DESAI, 2015).

Diversas soluções podem estar relacionadas a terceira camada, de aplicação, com a competência principal de englobar as múltiplas áreas demonstradas na Figura 1. Esta camada está relacionada as necessidades de usuário, portanto entrega o serviço final. Através de *User Interface* (UI) busca demonstrar as informações obtidas na camada de percepção ao usuário final, disponibilizando soluções de monitoramento e gestão através da Internet ou de aplicações locais (BILAL, 2017).

A última camada, de avaliação, está ligada a tomada de decisões inteligentes através dos dados gerados pelas outras camadas, desde as informações oferecidas pelo usuário na camada de rede até os dados gerados através da utilização das soluções desenvolvidas na camada de aplicação. Nesta camada são aplicadas técnicas de inteligência artificial e análise de dados (ARSéNIO, 2014).

A reunião destas camadas formula uma arquitetura heterogênea e interoperável, que propõe resolver problemas reais através da construção de soluções que influenciem nas diversas áreas da sociedade (RAY, 2016), auxiliando na solução dos desafios mundiais do futuro.

2.1.2 Arquitetura em serviços

Em *Wireless sensor network* (WSN) para aplicações de alta escala que possuam a transmissão de dados entre dispositivos embarcados e servidores locais ou remotos - IoT -, normalmente o número de nós é muito grande (ITU, 2012). Além disso, esta rede poderá possuir alterações dinâmicas, ou seja, os enlaces sem fio podem falhar com muita frequência ou terem alta substituição. Assim, o gerenciamento das informações provenientes dos sensores pode se tornar muito complexo (STANFORD-CLARK, 2013).

As camadas definidas para uma arquitetura IoT possuem serviços específicos, os quais são oferecidos por diversas ferramentas. A reunião desses serviços considera as particularidades das aplicações oferecidas nas camadas superiores, e define uma arquitetura orientada a eventos, com abordagens de centralização de atividades. Na Figura 4 podemos visualizar os serviços realizados na camada de rede, que é a mais importante para o funcionamento da aplicação.

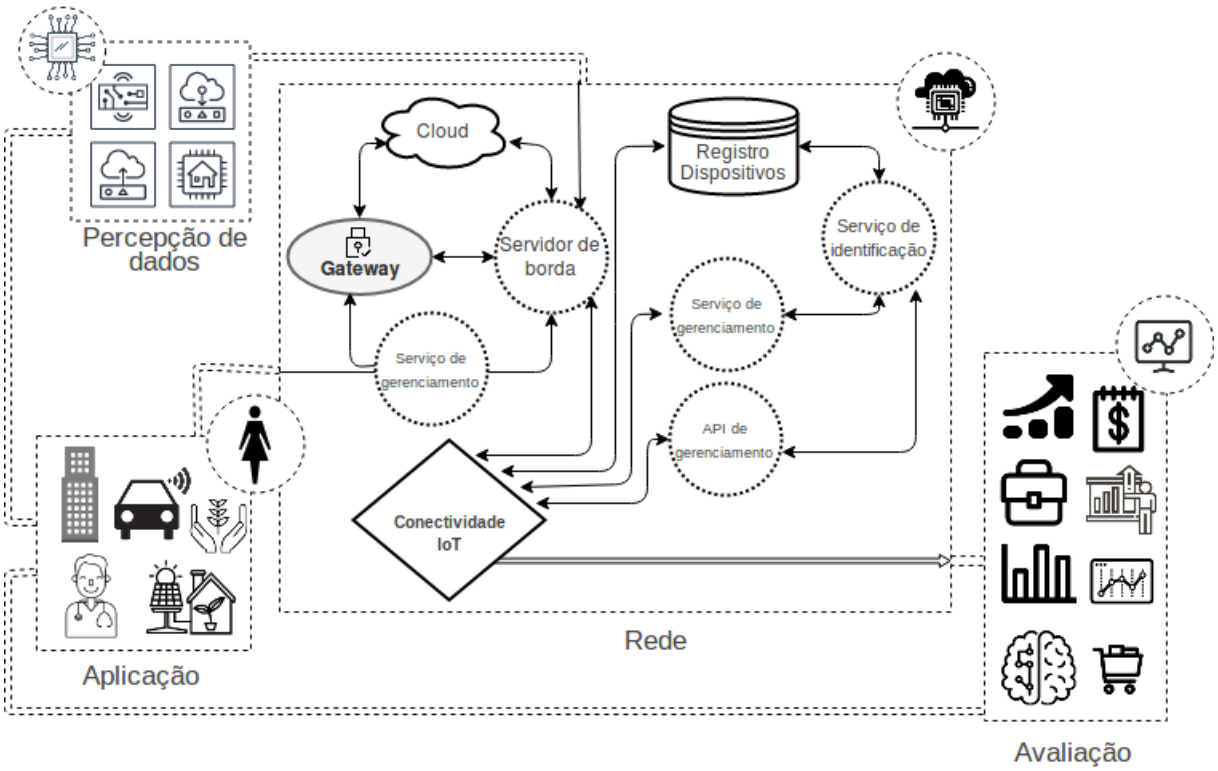

Figura 3 – Arquitetura IoT - Comunicação. *Fonte: Próprio autor, baseado em (IBM,).*

A camada de rede possui serviços que proporcionam a comunicação entre as camadas de percepção de dados, aplicação e avaliação. Seus módulos principais incluem servidores de borda, serviços de gerenciamento, *Cloud*, *Gateway*, API de gerenciamento, serviços de identificação e registro de dispositivos. Ou seja, gerencia, modulariza e armazena os dados provenientes das outras camadas da arquitetura.

As arquiteturas de aplicação que consideram abordagens para resolução de problemas de comunicação utilizam, em sua maioria, a comunicação centrada em dados. Através de filas de mensagens, realizam o gerenciamento das informações, com abordagem padrão para troca de mensagens, para possibilitar escalabilidade e uma topologia de rede dinâmica (STANFORD-CLARK, 2013).

Um serviço de gerenciamento normalmente utiliza uma fila de mensagens com o propósito de comunicação, de forma assíncrona, ou seja, sem tempo determinado, realiza o direcionamento de dados entre serviços. São utilizadas para armazenar informações de serviços dissociados, normalmente característicos em arquiteturas de microsserviços, disponibilizando recursos de comunicação e coordenação de aplicativos/serviços (COULOURIS, 2013).

Os servidores de borda realizam a tradução do acesso para a *Cloud*, e a conexão com o bloco de conectividade IoT. O serviço de conectividade interliga os dados captados pela camada de percepção com os serviços de armazenamento e gerenciamento da arquitetura, assim proporciona a tomada de decisões das diversas aplicações.

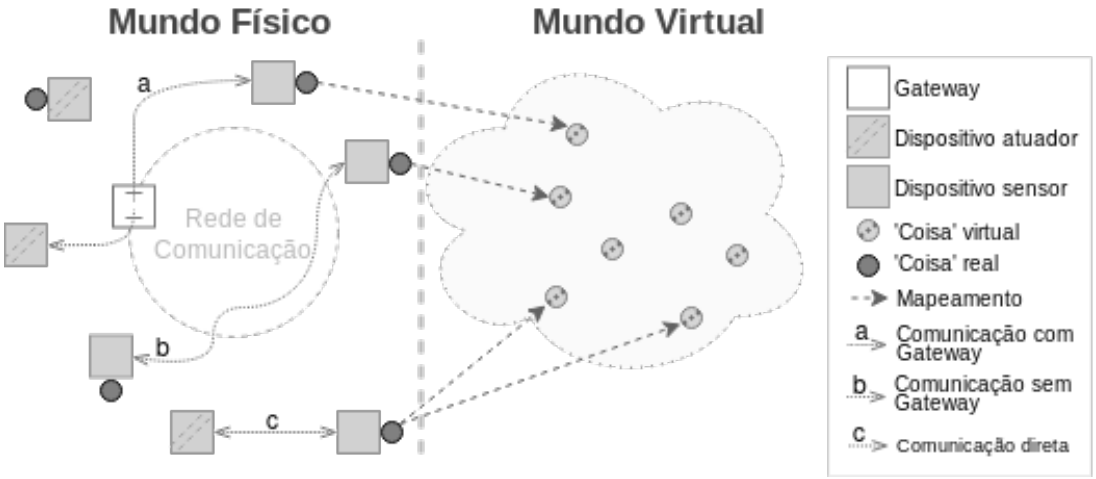

Figura 4 – Arquitetura IoT - Visão específica. *Fonte: Próprio autor, baseado em (ITU, 2012).*

A comunicação centrada em dados exige a utilização de diversos serviços, neste caso o *Gateway* tem o papel de direcionar as informações. Conforme podemos ver na Figura 4 o desenvolvimento baseado em diferentes modulações poderá criar uma aplicação não orientada a dados.

O cenário **a** indica a comunicação orientada a dados e normalmente persistente na Internet. o cenário **b** indica uma comunicação sem *Gateway*, o que possivelmente apenas transmite a informação sem tomada de decisões inteligentes. Por fim, o cenário **c** caracteriza uma comunicação M2M, ou seja, apenas há tomada de decisão direta na camada de aplicação.

2.1.3 Arquitetura de Software

Ao iniciar o desenvolvimento de uma aplicação, a definição da arquitetura é essencial para compreensão do funcionamento de cada módulo do sistema. O entendimento total do projeto possibilita escalabilidade, portabilidade de código, realização rápida de modificações pontuais além de facilitar a organização do projeto e trabalho em equipe. As arquiteturas de software podem ser definidas em cinco tipos diferentes - camadas, *Microkernel*, baseada em espaço, orientada a eventos e microsserviços -, que definem parâmetros para a qualidade de arquitetura e desenvolvimento de software, possuindo especificidades relativas a implementação de diferentes aplicações (RICHARDS, 2015).

A arquitetura em camadas normalmente divide a aplicação nas camadas principais de apresentação, negócios, persistência e banco de dados, com fácil distribuição de papéis, formula uma arquitetura sustentável e testável. A arquitetura de *Microkernel* possui como característica principal a separação em produtos, assim fornece recursos de isolamento em serviços e possibilita adição de recursos adicionais como *plug-ins*. Aplicações com possibilidade de alto fluxo de dados exigem uma arquitetura baseada em espaço, pois evita o colapso funcional sob alta carga, através da divisão de processamento e armazenamento entre vários servidores (RICHARDS, 2015).

A utilização de uma unidade central que delega atividades a módulos responsáveis com base no acontecimento de eventos, caracteriza a arquitetura orientada a eventos, que proporciona uma solução facilmente adaptável para ambientes complexos, com alta escalabilidade e extensibilidade (RICHARDS, 2015). Outra formulação de arquitetura é microsserviços, a qual possui alta divisão de serviços, ou seja, pequenas aplicações são utilizadas em conjunto para resolver problemas de escalabilidade e disponibilidade de serviço (RICHARDSON, 2018).

Arquiteturas para aplicações de IoT possuem a premissa principal de análise de eventos, portando a definição que mais adequada para este cenário é a orientada a eventos. No entanto, as aplicações podem sofrer constantes modificações com o surgimento de novas necessidades do usuário, o que adequaria o cenário a uma arquitetura de microsserviços. Assim sendo, o mais adequado para esta arquitetura é a utilização de definições mistas entre os dois padrões de arquitetura. As escolha de uma arquitetura mista é possível, porém as decisões de definição de arquitetura serão resolvidas no capítulo posterior, de Desenvolvimento.

2.2 Arquitetura Web

2.2.1 Formatação

As principais tecnologias utilizadas para a formatação de páginas e aplicativos Web são o *Hypertext Markup Language* (HTML) e o *Cascading Style Sheets* (CSS), que definem a estrutura, semântica, organização, *design* e realizam integração com serviços remotos. Essas ferramentas disponibilizam ao usuário a capacidade de realizar tarefas como publicação de documentos *online*, visualização de mídia e compartilhamento de informação (W3C, 2017a).

A linguagem de marcação de texto HTML é definida pela *World Wide Web* (W3), e possui como característica principal a capacidade de definir semanticamente objetos de interação com o usuário, como menus de opção, mídia, gráficos e estruturas de documentos. Assim, através de documentos *Standard Generalized Markup Language* (SGML) proporciona representação de informações em uma larga escala de domínios (W3C, 2017a).

A formatação do HTML é desempenhada por uma hierarquia de elementos, por meio de documentos separados em declaração, prólogo e instância (W3C, 2017b). Proporciona a formatação desde documentos estáticos a aplicativos dinâmicos, através do uso de marcações e uma sintaxe específica. Define os diversos objetos utilizados para compor aplicações Web, que podem ser utilizadas por diversos usuários, em locais diferentes e com baixos requisitos de processamento (W3C, 2017a).

```
1 <!DOCTYPE html>
2 <html>
3   <body>
4     <h2>The Button Element</h2>
5     <button class="moreinfo" type="button" onclick="alert('Hello world!')">Click Me!</button>
6   </body>
7 </html>
```

O desenvolvimento de aplicações com aparência agradável é necessário para que haja satisfação do usuário. O CSS foi desenvolvido pelo W3 para auxiliar os desenvolvedores a criar rapidamente aplicações Web, com boa aparência para Web-sites responsivos (W3C, 2018), definindo cores, leiaute e fontes (FLANAGAN, 2016).

A implementação e utilização do CSS possui requisitos como suporte a exibição do HTML semântico (W3C, 2016), realização de boas práticas de acessibilidade, além de permitir uma variedade de scripts internacionalizados (W3C, 2016) possibilitando a separação da estrutura da apresentação (FLANAGAN, 2016). Os padrões de CSS definidos pela W3 demonstram como definir propriedades, valores e regras para a demonstração gráfica dos objetos HTML.

O CSS define *classes* e atributos para a manipulação estética da página Web marcando elementos semanticamente. As classes do CSS definem atributos para uso em objetos do HTML. Os atributos do CSS, estão contidos nas classes, e quando utilizados atribuem valor estético ao objeto HTML por exemplo definindo cores, espaçamento, formatação de textos e imagens.

```
8 p.moreinfo { font-style: italic }
```

2.2.2 Endereçamento

Uma URI proporciona uma extensão simples para identificação de um recurso disponível na Internet, ou seja, reúne o nome do recurso *Uniform Resource Name* (URN) com a localização do recurso

Uniform Resource Locator (URL) para possibilitar ao usuário acesso distinto a aplicações na Web através dos padrões e semântica definidos pela iniciativa W3 (GROUP,).

As características relacionadas a uma URI padronizam sua criação e utilização. O aspecto de uniformidade possibilita definir novos recursos sem interferir na maneira como os identificadores existentes funcionam. A propriedade de disponibilidade de recurso, indica que uma URI pode fornecer identificação para diferentes componentes, como por exemplo uma imagem, uma página Web, um documento eletrônico e entre outros. Além disso, o padrão possui um identificador que define cada componente, ou seja, difere o acesso entre diversos componentes (BERNERS-LEE, 2005).

A sintaxe definida pela W3 relaciona os componentes para expor os recursos de forma única, a Figura 5 demonstra um exemplo de uma URI. Os aspectos esquema (*scheme*), autorização (*authority*), caminho (*path*), inquerimento (*query*) e o fragmento (*fragment*) compõem a indentificação. Através do qual, com a reunião desses componentes, constitui uma URI válida e que respeita as premissas de funcionamento independente, disponibilidade de serviço ao usuário e facilidade de identificação de recurso.

```
  foo://example.com:8042/over/there?name=ferret#nose
  \_/   \______________/\_________/ \_________/ \__/
   |           |            |            |        |
scheme     authority       path        query   fragment
   |   _____________________|__
  / \ /                        \
  urn:example:animal:ferret:nose
```

Figura 5 – Exemplo URI. *Fonte: Berners-Lee, Fielding e Masinter (2005).*

O *esquema* especifica as atribuições do recurso definidas pela *Best Current Practice* (BCP35) (THALER, 2005), a *autorização* define hierarquia e autoridade. O *caminho* entrega o resultado dos recursos disponíveis no escopo do esquema do URI. A componente *inquerimento* identifica o recurso solicitado, e o *fragmento* permite identificação indireta de recursos secundário.

2.2.3 Transferência de dados

O protocolo *Hypertext Transfer Protocol* (HTTP) é um protocolo cliente/servidor que trabalha na camada de aplicação para sistemas distribuídos e colaborativos através da transferência de hipermídia (GRIGORIKE, 2013), ou seja, realiza a transferência de dados na Internet independente do tipo de dado (NETWORK; CONTRIBUTORS, 2005–2018).

Este protocolo sem estado é simples e possibilita o controle de dados para diversas aplicações, que utilizem, por exemplo, sensores e atuadores para monitorar atividades cotidianas (NETWORK; CONTRIBUTORS, 2005–2018). Iniciou como um protocolo simples, de uma linha, para recuperar hipertexto. No entanto, diferentes utilizações exigem distintas características de resposta do protocolo, o que influenciou ao longo do tempo na realização de estudos e melhorias. (GRIGORIKE, 2013).

Devido as necessidades divergentes e novos desafios das aplicações modernas, a entidade W3 realizou diversas mudanças ao longo da história (GRIGORIKE, 2013). A primeira versão HTTP/0.9 é um protocolo cliente/servidor, que trabalha de forma simples baseada em caracteres ASCII e projetado para transferir HTML sem informações de cabeçalho.

A segunda versão, HTTP/1.0, é um pouco mais robusta, incluindo a notação de cabeçalhos ao protocolo e propiciando flexibilidade e extensibilidade a troca de dados, pois o cabeçalho *Content-Type*

permite transmitir outros tipos de documento, onde a versão HTTP/0.9 só aceitava HTML (NETWORK; CONTRIBUTORS, 2005–2018).

A versão mais atualizada deste protocolo é a HTTP/2.0, criada pelo grupo *HTTP Group Work* (HTTPbis), com o objetivo de obter melhores taxas de latência e desempenho devido as suas características de otimização e multiplexação (GRIGORIKE, 2013): binário, compressão de cabeçalhos e multiplexação de fluxos de solicitações e respostas (*streams*), este último visando resolver os problemas de *head-on-line blocking* de solicitações em fila (NETWORK; CONTRIBUTORS, 2005–2018).

2.2.4 *WebSocket*

O *WebSocket* é um protocolo que fornece serviços de interoperabilidade com infraestruturas Web através de uma API. Permitindo fluxo de dados *full-duplex* com servidores remotos (WEBSOCKET.ORG, 2018), para arquiteturas cliente/servidor. Realiza transporte de informações disponíveis no navegador Web, como por exemplo cargas simples como *JavaScript Object Notation* (JSON) ou mais complexas como formatos de mensagens binários mais complexos (GRIGORIKE, 2013).

Este protocolo, responsável pelo canal de comunicação, possibilita a utilização de *proxies* e aceita *firewalls*, assim habilita transmissão de *streaming*, através de HTTP, e possibilita conexões simultâneas(WEBSOCKET.ORG, 2018). As utilidades de comunicação do protocolo possibilitam esquemas próprios de URL, *WS* e *WSS*, para envio e recebimento de texto e dados binários em textos simples e através de canal criptografado (GRIGORIKE, 2013).

A comunicação cliente/servidor se da através de um canal estabelecido a partir da detecção de um servidor, podendo utilizar um *proxy*, definido pela URL (ORGANIZATION, 2012), e estabelecendo a emissão de uma instrução *HTTP CONNECT*. O servidor realiza uma conexão *Transmission Control Protocol* (TCP)/*Internet Protocol* (IP) com o host remoto estabelecido no cabeçalho da comunicação. Assim, realiza solicitações ao servidor para alterar o protocolo de transmissão de dados HTTP para *WebSocket*, pois por possuírem compatibilidade resolvem o canal bidirecional mesmo que hajam *proxies* no caminho (WEBSOCKET.ORG, 2018).

```
9  GET ws://echo.websocket.org/?encoding=text HTTP/1.1
10 Origin: http://websocket.org
11 Cookie: __utma=99as
12 Connection: Upgrade
13 Host: echo.websocket.org
14 Sec-WebSocket-Key: uRovscZjNol/umbTt5uKmw==
15 Upgrade: websocket
16 Sec-WebSocket-Version: 13
```

Caso o servidor tenha disponibilidade para uso do protocolo *WebSocket*, a resposta é dada através do cabeçalho *upgrade*.

```
17 HTTP/1.1 101 WebSocket Protocol Handshake
18 Date: Fri, 10 Feb 2012 17:38:18 GMT
19 Connection: Upgrade
20 Server: Kaazing Gateway
21 Upgrade: WebSocket
22 Access-Control-Allow-Origin: http://websocket.org
23 Access-Control-Allow-Credentials: true
24 Sec-WebSocket-Accept: rLHCkw/SKsO9GAH/ZSFhBATDKrU=
25 Access-Control-Allow-Headers: content-type
```

2.2.5 WebRTC

A transmissão de dados de *streaming* em tempo real, através da Internet pode se tornar ineficiente, se as ferramentas utilizadas são inadequadas (GRIGORIKE, 2013). O projeto aberto *WebRTC* visa criar e otimizar recursos para realização de comunicação em tempo real - *Real Time Communications* (RTC) através de APIs e protocolos de fácil utilização e entendimento (PROJECT, 2018).

As funcionalidades comuns de utilização do *WebRTC* incluem definição da conexão entre os pares que realizam a comunicação RTC. Esta conexão pode ser realizada através de tecnologias de *Network Address Translation* (NAT), com a utilização de APIs de uso específico para cada recurso (WEBRTC, 2018), que possibilitam as funcionalidades de sinalização, descoberta de *peers* (pontos), negociação de conexão e segurança (GRIGORIKE, 2013). Apesar desse conjunto de tecnologias proporcionar topologia ponto-a-ponto, por se tratar de utilização em RTC e ter influência com as características de proteção da rede como NAT, *Gateway*, e *firewalls* a comunicação necessita de *fallback* para a verificação de falha de conexão. Para realizar estas e outras funcionalidades, diversos serviços são acoplados a arquitetura, por exemplo os serviços de estabelecimento e manutenção de conexão (DEVELOPERS, 2018).

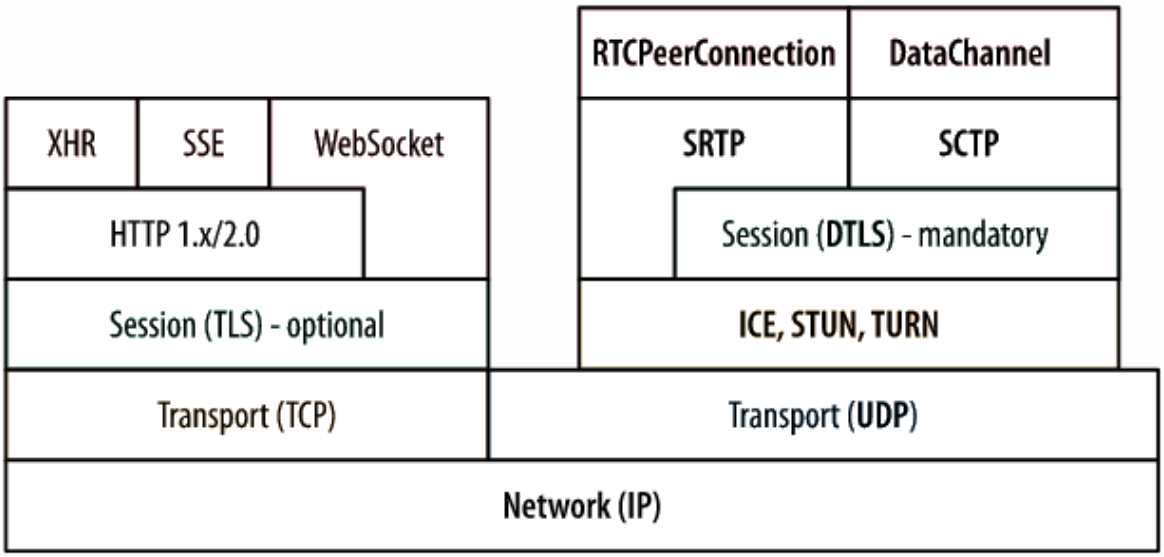

Figura 6 – Arquitetura *WebRTC Fonte: Grigorike (2013).*

2.3 Laboratórios remotos

Os laboratórios remotos são ferramentas utilizadas para oferecer acesso a experimentos e laboratórios de forma remota, ou seja, através da Internet (LOURENçO, 2018). O experimento analisado através da Web é real, e utilizado para aumentar a percepção sobre dados reais e medições em tempo real (RTC) (GRACILIANO, 2018), efetuadas a partir de sensores.

O papel destas ferramentas é possibilitar aos alunos o acesso ao experimento, com flexibilidade, independente de hora, local ou estrutura física, isto é, não sendo necessário grandes recursos computacionais para acesso e utilização dos experimentos reais. Porém, existem diversos desafios relacionados a disponibilização de laboratórios remotos.

Os desafios mais comuns estão relacionados a modulação do experimento, desenvolvimento de *software* e *hardware*, infraestrutura de rede e questões administrativas. Além de outros desafios como integração do sistema com o sistema acadêmico para controle de acessos, integração de materiais didáticos, controle disponibilidade de experimento (agendamento) (LOURENçO, 2018) e fácil manutenção.

A modularidade correta de uma arquitetura relacionada a este serviço poderá solucionar a maioria dos desafios relativos ao projeto, a utilização de uma infraestrutura para armazenamento dos experimentos facilita a manutenção e inserção de novos experimentos a grade curricular, além de considerar as diferentes soluções que podem ser incluídas na camada de aplicação.

Os diferentes experimentos que podem ser desenvolvidos em laboratórios remotos tem a característica comum de mudança de estado físico em sua estrutura educacional. Os sensores são dispositivos físicos que respondem a estímulos reais, como temperatura, umidade e pressão, portanto medem alterações físicas no experimento. Os atuadores são dispositivos que proporcionam acionamento de alguma funcionalidade do sistema, normalmente provenientes da ação do usuário. A utilização destes dispositivos físicos proporciona a realização de experimentos reais.

A verificação de alterações físicas de forma visual é crucial para o entendimento de todas as características dos experimentos, para isso é importante que a transmissão de dados de vídeo seja realizada. No entanto, é importante que as estruturas prevejam o funcionamento não ideal das redes de transmissão de dados, prevendo soluções que estejam de acordo com altas taxas de perca de dados.

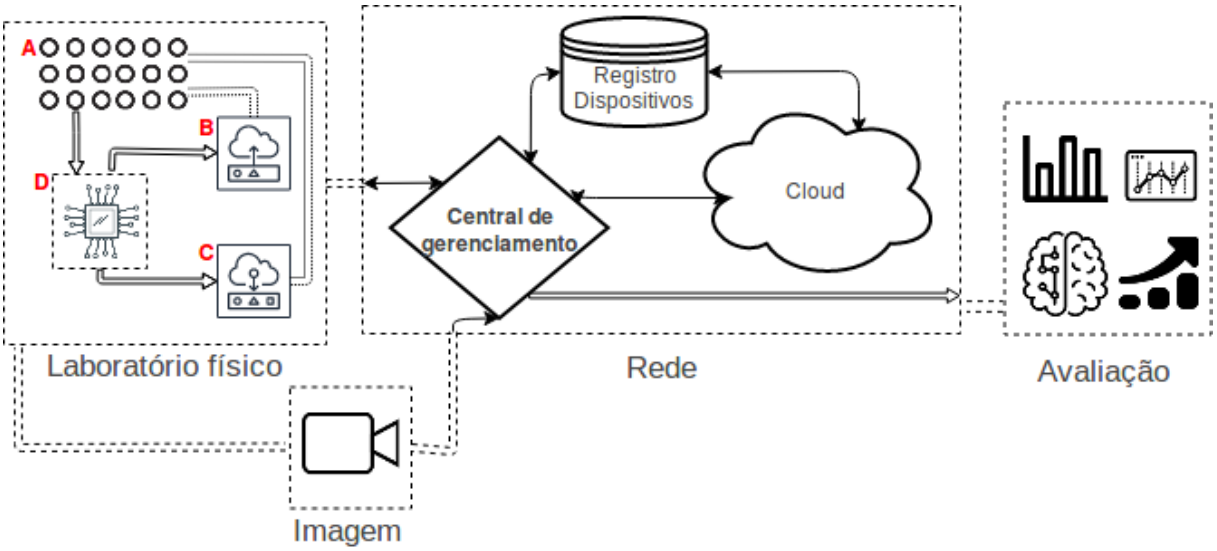

Figura 7 – Estrutura de laboratório remoto. *Fonte: Próprio autor baseado em VISIR (2018).*

Características como escalabilidade, heterogeneidade de hardware, interoperabilidade estão ligadas a infraestruturas de laboratórios remotos. Além disso, a inclusão de novos serviços e fácil manutenção devem ser consideradas para o desenvolvimento. Assim, é perceptível que é possível enquadrar uma infraestrutura de laboratórios remotos a uma infraestrutura de IoT.

A estrutura de laboratórios remotos pode ser analisada em quatro blocos principais, conforme podemos ver na Figura 7. O bloco de estrutura física (laboratório físico) que possui os experimentos multidisciplinares, o bloco de rede que possibilita a utilização do experimento através da Web e o bloco de avaliação que possibilita a análise dos dados gerados a partir dos experimentos e o bloco de imagem, que é facultativo para transmissão de imagem.

O laboratório físico possui como premissa a utilização de componentes multidisciplinares (**A**), que em conjunto com sensores (**B**) e atuadores (**C**), são normalmente monitorados por um hardware embarcado (**D**). Além disso, o bloco de Rede possui sub-blocos que possibilitam acesso aos dados (*Cloud*), armazenamento dos dados (*Registro dispositivos*) e demonstração de dados e imagem (*Central de gerenciamento*). Por fim, o bloco de rede é responsável pela analise dos resultados dos utilizadores dos experimentos, realizada por parte das instituições, devendo ser considerada a integração do sistema com os sistemas acadêmicos.

3 PROPOSTA

O desenvolvimento deste projeto propõe a implementação de uma infraestrutura para acesso, monitoramento e visualização de experimentos remotos. Esta arquitetura possui modulação em microsserviços e é orientada a eventos. Para a implementação de uma aplicação assíncrona, com controle de dados, interoperável e com diversidade de hardware a arquitetura de microsserviços será aplicada. As sessões seguintes demonstram os cenários identificados para esta aplicação, e que portanto, que serão desenvolvidos para este projeto.

3.1 Cenário 1 - Medição de dados

Esse cenário considera as alterações no valor da medição do sensor no sistema embarcado (**B**). Quando (**B**) identifica uma alteração de medição física, ou seja, um novo valor no sensor, uma requisição HTTP é enviada para a *Representational State Transfer* (REST) API (**E**) que controla o fluxo de informações entre as medições físicas e a demonstração em uma aplicação Web. Por fim, os dados são armazenados em um banco de dados não relacional (**C**), mantendo o registro das informações, e uma aplicação Web (**A**) demonstra os dados provenientes das medições físicas.

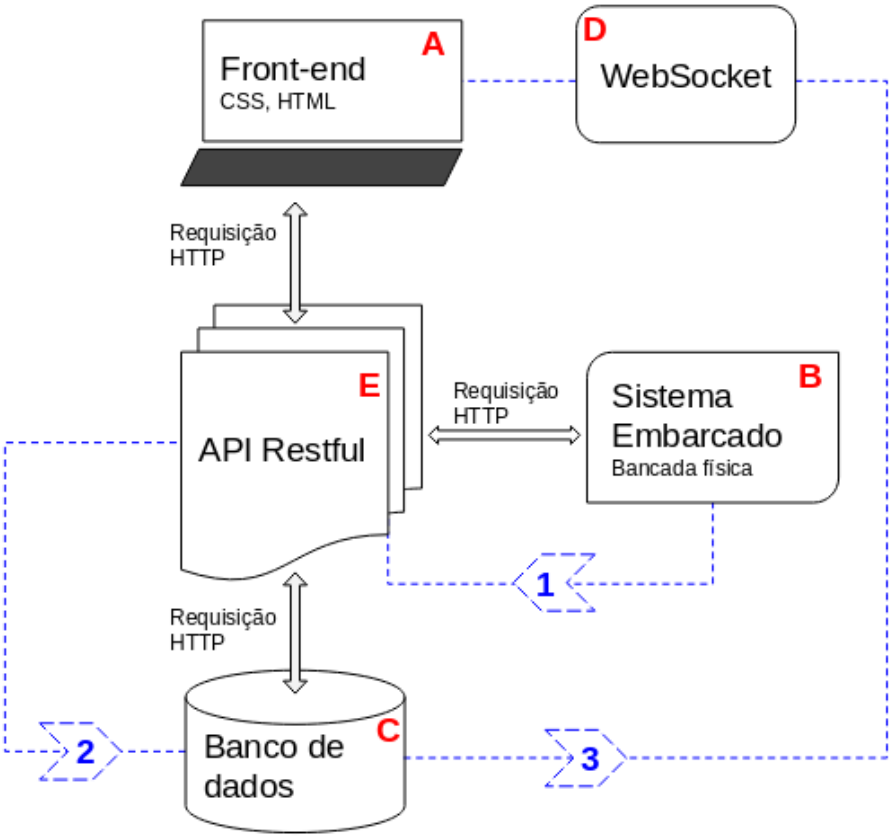

Figura 8 – Cenário 1 - Arquitetura proposta. *Fonte: Próprio autor.*

3.2 Cenário 2 - Atuação no sistema

Esse cenário considera as alterações em atuadores monitorados na página Web (**A**). Quando o usuário da aplicação Web (**A**) modifica o valor relativo a um atuador, uma requisição HTTP é enviada para a REST API (**E**), as informações são armazenadas no banco de dados não relacional (**C**). A API (**E**) envia os dados para um *Gateway* (**F**) que controla o fluxo de dados e transmite as mensagens para um *Broker* (**G**), que funciona como um servidor de mensagens, enviando as informações para o sistema embarcado (**B**).

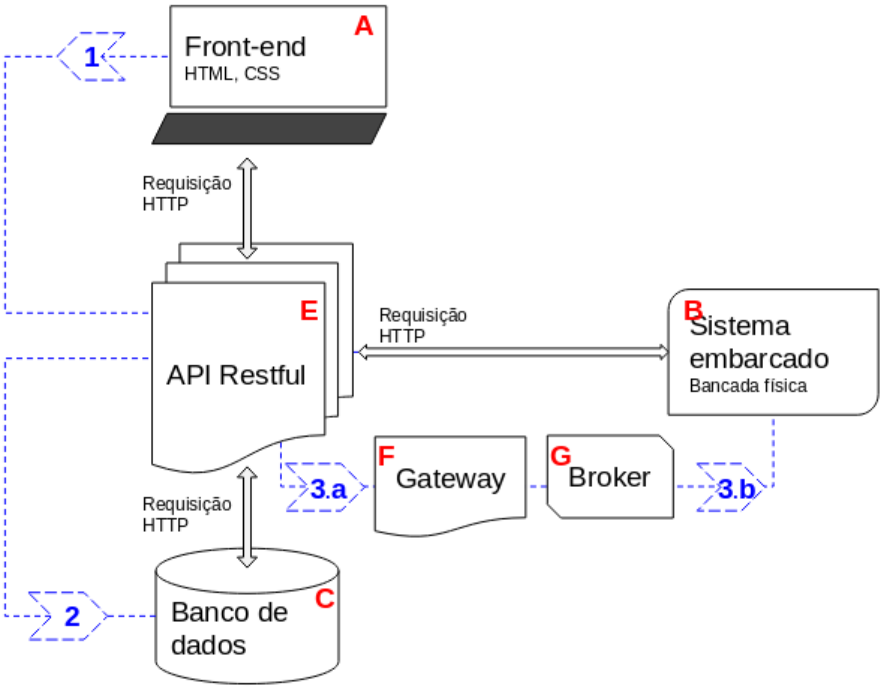

Figura 9 – Cenário 2 - Arquitetura proposta. *Fonte: Próprio autor.*

O experimento proposto para a validação da infraestrutura, possui o intuito principal de demonstrar as alterações de movimento de um motor compressor através da modificação da frequência de funcionamento de uma lâmpada estroboscópica. É possível visualizar o experimento real através de uma câmera IP, além de realizar o controle de sensores e atuadores através de acesso remoto gerenciado por uma REST API. Desta forma o estudante pode observar que experimentos reais são sensíveis a problemas reais, como falta de luz, variações de temperatura e influência de meio industrial, criando uma visão que auxiliará no desenvolvimento de projetos inerentes a atividades técnicas.

REFERÊNCIAS

ARSéNIO, A. M. Internet of intelligent things: Bringing artificial intelligence into things and communication networks. 10.1007/978-3-642-35016-0$_1$, 2014. *Citadonapágina*19.

BERNERS-LEE, T. *Uniform Resource Identifier (URI): Generic Syntax*. [S.l.], 2005. Disponível em: <https://www.w3.org/Addressing/URL/url-spec.txt>. Citado na página 23.

BERNERS-LEE, T.; FIELDING, R. T.; MASINTER, L. *Uniform Resource Identifier (URI): Generic Syntax*. [S.l.], 2005. <http://www.rfc-editor.org/rfc/rfc3986.txt>. Disponível em: <http://www.rfc-editor.org/rfc/rfc3986.txt>. Citado 2 vezes nas páginas 3 e 23.

BILAL, M. A review of internet of things architecture, technologies and analysis smartphone-based attacks against 3d printers. *CoRR*, abs/1708.04560, 2017. Disponível em: <http://arxiv.org/abs/1708.04560>. Citado 3 vezes nas páginas 3, 18 e 19.

BRANCO, M. *TICs para el aprendizaje de la Ingeniería*. Alfonso Lago Ferreiro, 2017. Disponível em: <http://romulo.det.uvigo.es/ticai/libros/2017/2017/Cap16.pdf>. Citado na página 13.

CIEB, C. de inovação para a educação brasileira. *POLÍTICAS DE TECNOLOGIA NA EDUCAÇÃO BRASILEIRA*. 2018. Disponível em: <http://www.cieb.net.br/wp-content/uploads/2017/11/CIEB-Estudos-4-Politicas-de-Tecnologia-na-Educacao-Brasileira.pdf>. Citado na página 14.

COULOURIS, G. *Sistemas Distribuídos - Conceitos e Projetos, 5 edição*. [S.l.]: Bookman, 2013. Citado na página 20.

DAS, D. J. C. S. K. *Smart Environments: Technologies, Protocols, and Applications*. [S.l.]: John Wiley Sons, Inc, 2005. Citado na página 18.

DESAI, A. Internet of things: Architecture and research challenges. *CSI Communications*, 2015 CSI, 2015. Citado 3 vezes nas páginas 3, 17 e 18.

DEVELOPERS, G. *Real time communication with WebRTC*. 2018. Disponível em: <https://codelabs.developers.google.com/codelabs/webrtc-web/#0>. Citado na página 25.

DINIZ, S. N. de F. *O uso das novas tecnologias em sala de aula*. 2001. Disponível em: <http://www.pucrs.br/ciencias/viali/doutorado/ptic/aulas/aula_2/187071.pdf>. Citado na página 13.

FLANAGAN, H. *CSS Requirements for RFCs - draft-iab-rfc-css-01*. 2016. Disponível em: <https://www.w3.org/TR/css-2018/>. Citado na página 22.

GAITAN, C. N. Gradual development of an iot architecture for real - world things. *IETE Technical Review*, IEEE, 2015. Citado na página 18.

GRACILIANO, J. *Laboratórios Remotos em Refrigeração*. 2018. Disponível em: <https://jesuegraciliano.wordpress.com/apresentacao-tecnologias-educacionais/laboratorios-remotos/>. Citado 2 vezes nas páginas 15 e 26.

GRIGORIKE, I. *High Performance Browser Networking*. [S.l.]: O'Reilly Media, 2013. Citado 4 vezes nas páginas 3, 23, 24 e 25.

GROUP, W. *URIs, URLs, and URNs: Clarifications and Recommendations 1.0*. Disponível em: <https://www.w3.org/TR/uri-clarification/>. Citado na página 23.

HöLLE, J. *From Machine-to-Machine to the Internet of Things - Introduction to a New Age of Intelligence*. [S.l.]: Elsevier Ltd, 2014. Citado 3 vezes nas páginas 3, 17 e 18.

IBM. *Internet of Things for insights from connected devices*. Disponível em: <https://www.ibm.com/cloud/garage/architectures/iotArchitecture/reference-architecture>. Citado 2 vezes nas páginas 3 e 19.

ITU, T. I. U. Overview of the internet of things. *SERIES Y: Global Informatio Infrastructure, internet protocol aspects and next-generation networks - Next Generation Networks – Frameworks and functional architecture models*, Am Diabetes Assoc, Y.2060, n. 2, 2012. Citado 6 vezes nas páginas 3, 14, 17, 18, 19 e 20.

LIMA, N. Do students really understand the difference between simulation and remote labs? *International Conference Technological Ecosystems for Enhancing Multiculturality 2017*, TEEM'17, v. 20, 2017. Citado na página 13.

LOURENçO, R. da S. *Laboratórios Remotos - Um Estudo para a PUC-Rio*. 2018. Disponível em: <http://www.puc-rio.br/pibic/relatorio_resumo2014/relatorios_pdf/ctc/ELE/ELE-RodrigodaSilvaLourenço.pdf>. Citado na página 26.

NETWORK, M. D.; CONTRIBUTORS individual. *HTTP Documentation - Creative Commons Attribution-ShareAlike*. 2005–2018. Disponível em: <https://www.w3.org/TR/css-2018/>. Citado 2 vezes nas páginas 23 e 24.

ORGANIZATION, W. *The WebSocket API*. [S.l.], 2012. <https://www.w3.org/TR/websockets/#terminology>. Disponível em: <https://www.w3.org/TR/websockets/#terminology>. Citado na página 24.

PORVIR. *TECNOLOGIA NA EDUCAÇÃO - Recomendações e experiências para transformar a maneira como se ensina e aprende a partir do uso de ferramentas digitais*. 2018. Disponível em: <http://porvir.org/especiais/tecnologia/#infraestrutura>. Citado na página 14.

PROJECT, W. *WebRTC Project*. 2018. Disponível em: <https://webrtc.org/>. Citado na página 25.

RAY, P. P. A survey on internet of things architectures. *Journal of King Saud University - Computer and Information Sciences*, Department of Computer Applications, Sikkim University, 2016. Citado na página 19.

RICHARDS, M. *Software Architecture Patterns - Understanding Common Architecture Patterns and When to Use Them*. [S.l.]: O'Reilly Media, 2015. Citado na página 21.

RICHARDSON, C. *Pattern: Microservice Architecture*. [S.l.], 2018. <https://microservices.io/patterns/microservices.html>. Disponível em: <https://microservices.io/patterns/microservices.html>. Citado na página 21.

STANFORD-CLARK, A. Mqtt for sensor networks (mqtt-sn) protocol specification. *International Business Machines Corporation (IBM)*, v. 1.2, 2013. Citado 2 vezes nas páginas 19 e 20.

THALER, E. D. *Uniform Resource Identifier (URI): Generic Syntax*. [S.l.], 2005. Disponível em: <https://tools.ietf.org/html/bcp35>. Citado na página 23.

(TPE), O. T. P. E. *INOVAÇÕES TECNOLÓGICAS NA EDUCAÇÃO: CONTRIBUIÇÕES PARA GESTORES PÚBLICOS*. 2018. Disponível em: <https://s3.amazonaws.com/porvir/wp-content/uploads/2014/08/Inovaç~oes-Tecnológicas-na-Educaç~ao_Constribuiç~oes-para-gestores-públicos-1.pdf>. Citado na página 14.

VISIR. *VISIR - RExLab*. 2018. Disponível em: <https://rexlab.ufsc.br/visir/>. Citado 2 vezes nas páginas 3 e 26.

W3C, W. G. N. *HTML CSS*. 2016. Disponível em: <https://www.w3.org/TR/css-2018/>. Citado na página 22.

W3C, W. G. N. *HTML 5.2*. 2017. Disponível em: <https://www.w3.org/TR/2017/REC-html52-20171214/introduction.html#introduction>. Citado na página 22.

W3C, W. G. N. *HTML 5.2*. 2017. Disponível em: <https://www.w3.org/TR/html52/>. Citado na página 22.

W3C, W. G. N. *CSS Snapshot 2018*. 2018. Disponível em: <https://www.w3.org/standards/webdesign/htmlcss>. Citado na página 22.

WEBRTC. *WebRTC 1.0: Real-time Communication Between Browsers*. 2018. Disponível em: <https://www.w3.org/TR/2018/CR-webrtc-20180927/>. Citado na página 25.

WEBSOCKET.ORG. *About HTML5 WebSocket*. 2018. Disponível em: <https://www.websocket.org/aboutwebsocket.html>. Citado na página 24.