

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET  
CÂMPUS GUARAPUAVA

NORTON MATIAS BINO DOS SANTOS

**APLICAÇÃO WEB PARA GERENCIAMENTO DE DISPOSITIVOS  
DE INTERNET DAS COISAS**

PROJETO DE TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA  
2018

NORTON MATIAS BINO DOS SANTOS

**APLICAÇÃO WEB PARA GERENCIAMENTO DE DISPOSITIVOS  
DE INTERNET DAS COISAS**

Projeto de Trabalho de Conclusão de Curso apresentado ao  
Câmpus Guarapuava da Universidade Tecnológica Federal do  
Paraná, como requisito parcial para a obtenção do título de  
Tecnólogo em Sistemas para Internet.

Orientador: Professor Me. Hermano Pereira  
UTFPR-GP

Coorientador: Professor Dr. Diego Marczal  
UTFPR-GP

GUARAPUAVA  
2018

## RESUMO

M B Santos, Norton. Aplicação Web Para Gerenciamento de Dispositivos de Internet das Coisas. 2018. 19 f. Projeto de Trabalho de Conclusão de Curso – Câmpus Guarapuava, Universidade Tecnológica Federal do Paraná. Guarapuava, 2018.

Ao longo dos anos, a segurança tem sido um tema muito comum e importante, desta forma as pessoas continuam buscando e desenvolvendo meios para se proteger de acidentes. Dito isto, é possível encontrar diversos produtos de segurança à venda no mercado, e alguns desses são dispositivos simples que emitem apenas uma notificação local e não possuem um histórico de notificações dos aparelhos. Com a chegada da Internet das Coisas, abre-se novas possibilidades, uma vez que dispositivos podem agora se comunicar pela Internet. Sendo assim, neste projeto é proposto o desenvolvimento de uma aplicação que resolva esses problemas apresentados, entregando uma interface gráfica para que o usuário possa consultar suas notificações e informações dos seus dispositivos.

**Palavras-chave:** Internet das Coisas. Ambiente Simulado. Web.

## ABSTRACT

M B Santos, Norton. Web Application For IoT Device Management. 2018. 19 f. Projeto de Trabalho de Conclusão de Curso – Câmpus Guarapuava, Universidade Tecnológica Federal do Paraná. Guarapuava, 2018.

Over the years safety has been a very common and important issue, so people continue to seek and develop ways to protect themselves from accidents. That said, it is possible to find several security products for sale in the market, and some of these are simple devices that can only send a local notification and do not have a history of notifications of the devices. With the advent of the Internet of Things, new possibilities open up, since devices can now communicate over the Internet. Therefore, in this project it is proposed the development of an application that solves these presented problems, providing an interface so that the user can consult their notifications and informations about their devices.

**Keywords:** Internet of Things. Simulated Environment. Web.

## LISTA DE FIGURAS

Figura 1 – Dispositivo Bluelux Pro . . . . .	5
Figura 2 – Simulador Cooja . . . . .	8
Figura 3 – Advanved Sky GUI . . . . .	9
Figura 4 – Quadro Kanban simples . . . . .	10
Figura 5 – Quadro Kanban do projeto no Trello . . . . .	11
Figura 6 – Arquitetura do sistema . . . . .	12
Figura 7 – Mockup - Dashboard aplicação . . . . .	13
Figura 8 – Mockup - Formulário de cadastro de dispositivos . . . . .	14
Figura 9 – Mockup - Tela de apresentação do dispositivo . . . . .	14
Figura 10 – Diagrama de Banco de dados . . . . .	15

## LISTA DE ABREVIATURAS E SIGLAS

IoT	<i>Internet of Things, ou Internet das Coisas</i>
HTTP	<i>HyperText Transfer Protocol</i>
API	<i>Application Programming Interface</i>
CoaP	<i>Constrained Application Protocol</i>
M2M	<i>Machine to Machine</i>
REST	<i>Representational State Transfer</i>
URL	<i>Uniform Resource Locator</i>
3G	<i>Third Generation</i>
WiFi	<i>Wireless Fidelity</i>
SQL	<i>Structured Query Language</i>
IP	<i>Internet Protocol</i>
HTML	<i>Hypertext Markup Language</i>

# SUMÁRIO

<b>1 – INTRODUÇÃO</b>	<b>1</b>
1.1 Objetivos	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
<b>2 – Aplicações Web IoT</b>	<b>3</b>
2.1 Protocolos Web para IoT	3
<b>3 – Trabalhos Correlatos</b>	<b>4</b>
3.1 BLUELUX	4
3.2 IBM DeveloperWorks tutorial	4
<b>4 – Aplicação Proposta</b>	<b>6</b>
4.1 Aplicação Web	6
4.2 Sensores	7
4.3 Simulador	7
4.3.1 Sensores no simulador	8
<b>5 – METODOLOGIA DE DESENVOLVIMENTO</b>	<b>10</b>
5.1 KANBAN	10
5.2 KANBAN APLICADO NO PROJETO	11
<b>6 – Desenvolvimento</b>	<b>12</b>
6.1 Arquitetura do Sistema	12
6.2 Protótipos de Tela	12
6.3 Diagrama de banco de dados	15
6.4 Histórias da Aplicação	15
6.5 Cronograma	16
<b>7 – RESULTADOS</b>	<b>17</b>
7.1 Resultados Parciais	17
7.2 Resultados Esperados	17
<b>8 – CONSIDERAÇÕES FINAIS</b>	<b>18</b>
<b>Referências</b>	<b>19</b>

## 1 INTRODUÇÃO

Com o passar dos anos a segurança tem sido um tema comum nas residências, sendo esta destinada tanto para as pessoas quanto para os bens que lá estão. Em relatório nacional, sobre acidentes e incidentes domésticos, publicado pelo [Inmetro \(2017\)](#), percebe-se uma redução de 290 casos, em 2016, para 176, em 2017, apesar de haver redução, ainda há a necessidade de aumentar a segurança residencial. Devido a demanda de segurança nas residências, é possível encontrar no mercado diversos produtos com tal finalidade, como por exemplo alarmes e sensores. Muitos desses produtos funcionam de forma independente, ou seja, sem a necessidade, ou possibilidade, de se comunicar com outros sistemas.

Pensando em segurança, é pertinente que a informação chegue até o responsável de forma prática, rápida e com o mínimo de ruído possível, sendo assim dispositivos de segurança independentes tornam-se ineficientes. Isso se deve ao fato do usuário não possuir um acesso facilitado aos seus dispositivos à longa distância. Além disso, pode não existir um histórico de acionamento e desligamentos dos dispositivos, ocasionando, em alguns casos, o não conhecimento do usuário sobre as notificações do sistema.

A Internet das Coisas (*Internet of Things* ou IoT) é uma rede de objetos físicos conectados entre si e com a Internet, capaz de coletar e transmitir dados. Esta possui as mais diversas finalidades, como a execução de ações ou até coleta dados para futuras tomadas de decisões.

*Smart Home* é um assunto crescente dentro da Internet das Coisas, em parte por conta da sua proposta de automação residencial e melhoria na qualidade de vida, possibilitando que o usuário verifique ou receba notificações do estado de sua residência e que possa comandar objetos à distância.

De acordo com o que se observa, este trabalho apresenta uma proposta para uma mudança na apresentação das notificações ao usuário, para que o mesmo possa sempre ser informado sobre a alteração de estados da sua residência. Também é interessante que as notificações sejam guardadas, de forma que possam ser consultados futuramente, e por fim seria interessante que alguns dispositivos pudessem ser desativados à distância, como por exemplo os alarmes.

A aplicação proposta será dividida em duas partes: uma *API* capaz de se comunicar com um simulador de dispositivos de IoT, que através de requisições *HTTP* possa cadastrar dispositivos e usuários, receber mensagens dos dispositivos e transformá-las em notificações para o usuário e permitir que alguns de seus módulos sejam desligados à distância; A segunda parte do projeto será uma aplicação *front-end* que se comunique com essa *API*.



## 1.1 Objetivos

### 1.1.1 Objetivo Geral

O objetivo geral deste trabalho é o desenvolvimento de um sistema *Web* capaz de se comunicar com um conjunto de dispositivos de IoT, previamente preparados, simulados no ambiente Cooja.

### 1.1.2 Objetivos Específicos

- Desenvolver os seguintes módulos *Web*: gerenciador de dispositivo e gerenciador de notificações.
- Implantar o protocolo de comunicação CoAP no servidor *Web* para comunicação com o ambiente simulado.
- Preparar um ambiente simulado com os dispositivos propostos.

## 2 Aplicações Web IoT

Do inglês Internet of Things (IoT), a Internet das Coisas refere-se à integração de objetos físicos e virtuais em redes conectadas à Internet, permitindo que “coisas” colem, troquem e armazenem uma enorme quantidade de dados numa nuvem, em que uma vez processados e analisados, gerem informações e serviços em escala inimaginável, [ALMEIDA \(2015\)](#). Através desta definição, é importante dizer que os dispositivos de Internet das Coisas muitas vezes não irão processar os dados, mas sim coletá-los. Dessa forma tornando interessante que esses dados sejam repassados e tratados em outro lugar. Para tratar os dados coletados pode-se utilizar uma aplicação para Internet, essa que estaria localizada em um servidor.

As aplicações *Web* para *IoT* muitas vezes possuem duas funcionalidades, a primeira sendo receber os dados e transformá-los em informação, podendo possuir uma interface gráfica simples apenas para verificação dos dados. Porém, também acontece de algumas aplicações possuírem interfaces gráficas melhores elaboradas, essas que entregam mais informações.

### 2.1 Protocolos Web para IoT

Dentre os protocolos Web para IoT o escolhido para ser trabalhado no projeto é o CoAP, *Constrained Application Protocol*. O *The Constrained Application Protocol* (CoAP) é um protocolo de transferência de dados da *Web* especializado para uso com nós restritos e redes restritas na Internet das coisas. O protocolo foi projetado para aplicações máquina a máquina (M2M), como energia inteligente e automação predial, segundo [Bormann \(2014\)](#). O CoAP se baseia no protocolo REST, onde o servidor libera serviços por *URL*, e clientes acessam esses recursos usando métodos como *GET*, *PUT*, *POST* e *DELETE*, também descrito por [Bormann \(2014\)](#).

### 3 Trabalhos Correlatos

Atualmente é possível encontrar algumas empresas que fornecem produtos com a finalidade de automatização, residencial e empresarial, e alguns projetos abertos com finalidades similares.

#### 3.1 BLUELUX

A Bluelux é uma empresa privada fundada em 2015, em Minas Gerais, que fornece seus próprios dispositivos e uma aplicação móvel para gerenciar lâmpadas e equipamentos eletrônicos, prometendo flexibilidade, segurança e custo-benefício. Seus módulos são *bivolt* e instalados entre o interruptor e a lâmpada, enviando um sinal Bluetooth, que seu aplicativo reconhece e cadastra o dispositivo na conta vinculada.

Ela possui três módulos principais, o Bluelux Socket, que funciona com uma lâmpada apenas, o Bluelux Pro, que permite a conexão de mais de uma lâmpada, mas possui um limite de *watts*, e o Bluelux Switch, que além das lâmpadas também permite conectar aparelhos eletrônicos diversos. Além dos dispositivos de IoT oferecidos pela empresa, a empresa também vende uma central de automação, que permite o controle dos seus módulos, dispensando a necessidade da utilização de *smartphones* pessoais, porém não é necessária para o funcionamento dos demais.

Atualmente, para utilização de seus produtos, está disponível um aplicativo móvel da empresa, sua única interface gráfica, que como já mencionado, cadastra os dispositivos próximos e permite a comunicação do usuário com o IoT. Esse aplicativo por sua vez possui uma interface amigável e fácil de ser utilizada e permite uma personalização no cadastro dos dispositivos, porém possui algumas funcionalidades reduzidas sem a central de automação.

Outras funcionalidades que a empresa entrega são a acessibilidade de qualquer lugar e o *Timeflow*, um recurso para controle de brilho da lâmpadas automatizado. Porém para o funcionamento do sistema à distância necessita de uma central de automação, um dispositivo Android, com conexão 3G ou Wi-Fi.

#### 3.2 IBM DeveloperWorks tutorial

A DeveloperWorks é um portal de acesso gratuito fornecido pela IBM para compartilhamento de tutoriais, ferramentas e códigos para ajudar a resolver problemas complexos. No tutorial postado por [Carneiro \(2014\)](#) é proposto o desenvolvimento de uma aplicação *Web* para coletar dados de um acelerômetro.

Esse projeto se utiliza de um dispositivo de IoT, Arduino Uno, juntamente de um sensor de aceleração para a coleta de dados, de uma aplicação *Web* em Ruby on Rails para disponibilização dos dados *online* e em tempo real, de um banco de dados SQL para



Figura 1 – Dispositivo Bluelux Pro

Fonte: [Bluelux \(2014\)](#)

armazenamento dos dados coletados e de um servidor Bluemix da IBM para manter a aplicação *Web online*. Por fim, o tutorial leva ao desenvolvimento de uma aplicação adaptativa capaz de se comunicar com uma placa programável Arduino uno e cita a possibilidade de utilização do seu projeto para outros temas e não apenas o acelerômetro.

## 4 Aplicação Proposta

Neste capítulo está descrita aplicação web proposta, apresenta-se as características que o sistema terá, os sensores a serem utilizados e o simulador escolhido.

### 4.1 Aplicação Web

Este projeto propõe uma solução para dois problemas de notificação que ocorrem nos dispositivos simples de segurança residencial. 1) O recebimento das notificações, uma vez que quando alguns dispositivos são acionados eles apenas geram uma notificação local, porém poderão existir casos em que o responsável não esteja próximo e não consiga verificar aquela notificação. 2) Além disso, muitas vezes não há um histórico de notificações, esses que possibilitariam o usuário saber quando uma notificação foi enviada e quantas notificações um dispositivo enviou.

A aplicação proposta será um sistema *Web* capaz de se comunicar com dispositivos de *IoT*. Também terá uma interface gráfica para que o usuário possa visualizar as notificações dos seus dispositivos, e que possibilite acessar históricos de ativação dos mesmos. Desta forma, a aplicação pretende auxiliar na segurança física de um ambiente.

O sistema irá prover o módulo do usuário, que permitirá o cadastro, a ser realizado pelo próprio, através de um formulário com os campos nome, *email* e senha. O usuário também poderá acessar o sistema com o seu *email* e senha, através de um formulário de *login*.

Da mesma forma, o sistema deverá entregar o módulo de Dispositivos, que será a representação dos aparelhos de Internet das Coisas. Esse modelo será composto pelos campos obrigatórios nome, categoria e Endereço de IP, e os campos não obrigatórios de descrição e de grupo, esse último que será uma referência. O dado Endereço de IP visa auxiliar a aplicação em localizar o dispositivo na Internet.

Do mesmo modo, os grupos poderão ser cadastrados pelo usuário, para que possa organizar seus dispositivos. Já os grupos conseguirão ser criados através de um formulário com os campos de nome e de descrição.

Contudo, no caso das notificações também é proposta a possibilidade de verificar o histórico das mesmas. Para isso as notificações devem ser salvas no banco de dados, e também deverá possuir as colunas identificador do dispositivo, mensagem e data e hora de criação da notificação.

Além disso, será possível verificar as notificações recebidas, que ainda não foram visualizadas, na página inicial, que possuirá uma *dashboard*, apresentando também os dispositivos e grupos do usuário. Tanto as notificações, os dispositivos e os grupos serão modelos, e deverão apresentar os métodos de criar, atualizar, mostrar e deletar.

O sistema será dividido em duas aplicações, uma que será responsável pelo *back-end*

e outra pelo *front-end*. A aplicação *back-end* será desenvolvida em Ruby on Rails. Rails é um *framework* de desenvolvimento de aplicações *Web* escrito na linguagem de programação Ruby, [Rails \(2007\)](#). A aplicação irá fornecer *endpoints*, e se utilizará do protocolo de comunicação CoAP, para permitir a troca de dados com os dispositivos. Desta forma, o sistema poderá consultar o estado dos dispositivos e os dispositivos podem enviar requisições com os dados da notificação e o sistema será responsável por guardar e apresentar a notificação ao usuário.

Além disso também possuirá uma aplicação *front-end*, que será desenvolvida em Angular. Como descrito por [Angular.io \(2018\)](#), o Angular é uma plataforma e *framework* para criação de aplicações em HTML e TypeScript. Contudo, para apresentar informações nas telas do sistema, essa aplicação deverá se comunicar com o *back-end* através do protocolo REST, utilizando os *endpoints* da aplicação.

## 4.2 Sensores

Segundo [BIPM \(2008\)](#), um sensor é uma parte de um sistema de medição, que é diretamente afetado por um fenômeno, corpo ou substância que possui uma quantidade a ser medida. Seguindo essa definição, os sensores que serão utilizados devem captar estímulos e, junto com o dispositivo, repassar a informação para o sistema.

Para o sistema proposto serão utilizados cinco tipos de sensores, sendo eles o de gases nocivos, um detector de fumaça, um detector de alagamento, um sensor de tranca e um alarme. Os detectores de gases nocivos e de fumaça serão sensores químicos, que quando entram em contato com um gás, ou fumaça, eles geram uma reação química que gera estímulos elétricos. Bem como, o sensor de alagamento, que funciona enviando um estímulo elétrico quando a água, que encosta no sensor, fecha o circuito elétrico. Por fim, o sensor da tranca irá validar o estado da fechadura, verificando se está trancada ou não.

Os sensores estarão vinculados aos dispositivos, sendo assim quando um dispositivo estiver cadastrado na aplicação, esse poderá enviar o estado do seu sensor para a aplicação.

## 4.3 Simulador

Devido ao foco do projeto ser uma aplicação para a Internet e não a construção de dispositivos de *IoT*, será utilizado um simulador para os dispositivos. Para isso serão desenvolvidos códigos na linguagem C que representarão, de forma simples, um dispositivo de *IoT*. O simulador em conjunto com os códigos e o protocolo de comunicação, tornarão possível a comunicação entre o sistema *web* e os dispositivos simulados.

O simulador que será utilizado é o Cooja, um programa desenvolvido na linguagem Java e que emula o sistema operacional Contiki. Com esse é possível gerar uma simulação, em que podem ser adicionados dispositivos. Para emular um dispositivo através desse programa é necessária a criação de um Makefile, que é um arquivo que define regras de compilação, e a

adição do código que defina as funcionalidades do dispositivo, desta forma o Cooja irá compilar o código e gerar a simulação.

Os dispositivos simulados irão gerar as notificações de acordo com os estímulos enviados pelos sensores, e salva-las pelos *end-points* da aplicação *Web*, através do método *POST*.

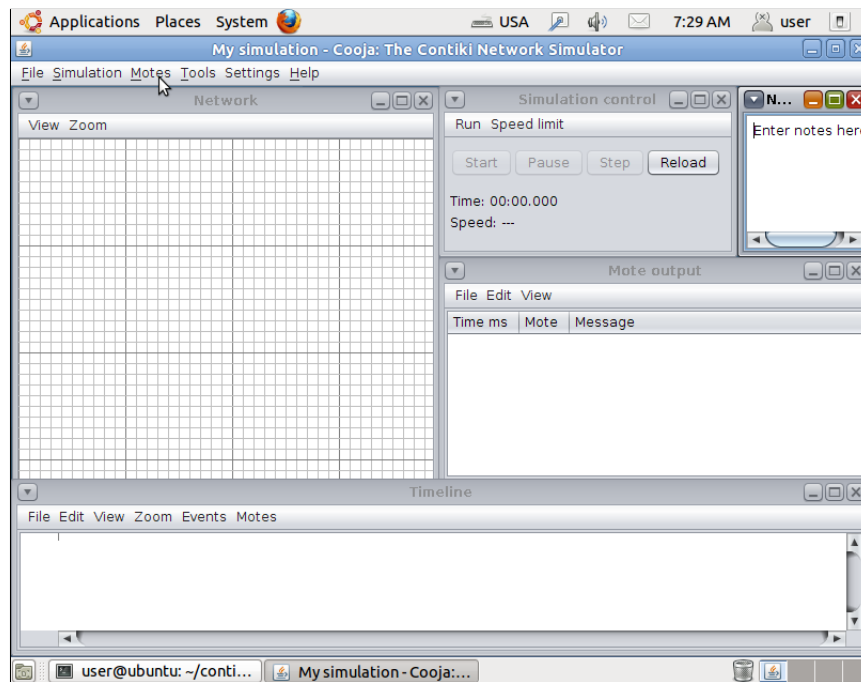


Figura 2 – Simulador Cooja

#### 4.3.1 Sensores no simulador

Os sensores também farão parte do ambiente simulado, e também serão simulados para a aplicação. Desta forma os dados que serão coletados e fornecidos poderão ser adicionados tanto de forma manual, aonde será inserido por uma interface gráfica, ou poderão ser gerados de maneira automática por bibliotecas.

Para um geramento automatizado de dados poderão ser usadas as bibliotecas *sht11-sensor.h*, estas desenvolvidas em C e entregam um formato de dados específico. Já para a entrega de dados manual será utilizada um plugin chamado *Advanced Sky GUI*, este que entrega uma interface gráfica e permite a inserção de dados



Figura 3 – Advanved Sky GUI



## 5 METODOLOGIA DE DESENVOLVIMENTO

Neste capítulo será abordada a metodologia ágil de desenvolvimento Kanban e como essa metodologia será utilizada no projeto.

### 5.1 KANBAN

O Kanban é uma ferramenta para gerenciar o fluxo de materiais ou informações, ou qualquer outra, em um processo, segundo definição de Klipp (2014). O nome está diretamente relacionado à forma de organização que essa metodologia propõe, Kanban significa “cartão” em japonês, que é um de seus principais aspectos. Essa metodologia foi originada na fábrica da Toyota, em 1960, como um sistema de controle de abastecimento e estoque. Ao longo do tempo também foi adaptada ao desenvolvimento de *software*, fazendo parte de outras metodologias ágeis para organizar o processo produtivo.

Essa metodologia possui duas principais características, sendo a primeira o quadro Kanban, uma ferramenta utilizada para auxiliar o desenvolvedor em acompanhar o andamento das etapas do projeto, assim como o desempenho da equipe. O quadro pode ser dividido em várias etapas, ou camadas, que a equipe deve seguir, sendo as mais comuns: *TODO*, para fazer, *IN PROGRESS*, em progresso, *CODE REVIEW*, revisão de código, e *DONE*, pronto.

Já a segunda característica são os cartões, eles representam histórias ou tarefas que devem ser realizadas. Em desenvolvimento de *software* o conjunto de histórias que possivelmente serão desenvolvidas é chamado de *backlog*, isso se deve ao fato que durante o desenvolvimento do projeto algumas histórias podem ser retiradas. Os cartões são passados entre as partes, ou etapas, do quadro sinalizando o andamento do projeto, sendo assim, um cartão que passar para revisão de código, significa que o desenvolvedor terminou o trabalho daquele cartão, e deixa liberado para ser revisado.

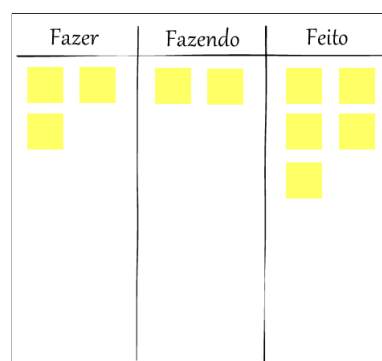


Figura 4 – Quadro Kanban simples

Fonte: [CulturaÁgil \(2014\)](#)

## 5.2 KANBAN APLICADO NO PROJETO

Para o projeto será utilizado o Trello, uma plataforma *web* de gerenciamento de quadros virtuais. Essa tecnologia permite a criação de listas dentro do quadro, e nas listas podem ser adicionados cartões, que representam as histórias ou tarefas. Os cartões serão escritos em formato de histórias, essas que são descrições em alto nível das funcionalidades do sistema, e demonstram como o usuário as enxerga. Quando uma história for enviada a desenvolvimento ela será quebrada em tarefas, que são descrições de menor nível, e mais detalhadas, para o desenvolvedor.

O projeto também possuirá *sprint*, que são períodos pré-determinados de tempo para a realização das tarefas, porém não será utilizada nenhuma outra metodologia de desenvolvimento além do Kanban. É proposto que as *sprint* tenham de duas semanas de duração, e as histórias que as compõe serão selecionadas no início de cada período pelo aluno.

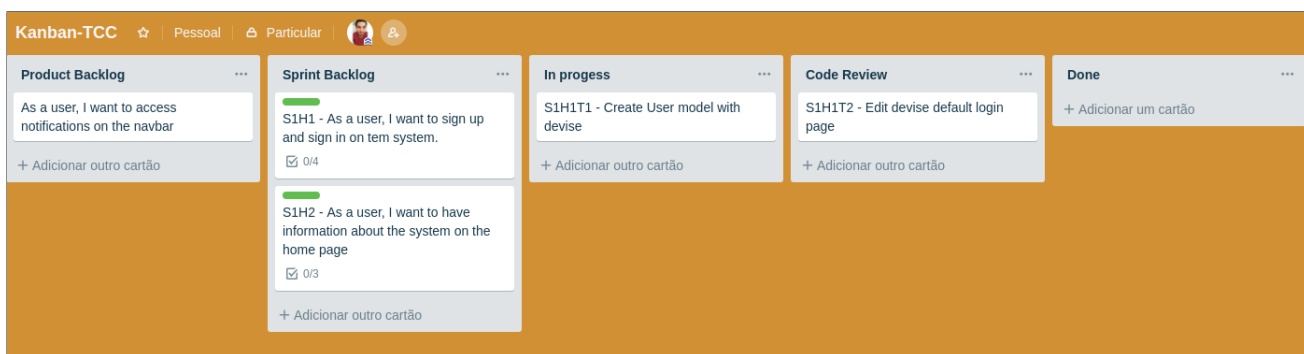


Figura 5 – Quadro Kanban do projeto no Trello

Fonte: [Atlassian \(2018\)](#)

No quadro do projeto serão utilizadas as seguintes listas: *Product Backlog*, sinalizando as histórias do projeto que possivelmente serão desenvolvidas, *Sprint Backlog*, sendo as histórias selecionadas para serem realizadas durante a *sprint* atual, *In Progress*, que estarão as histórias e tarefas que foram selecionadas para serem desenvolvidas, *REVIEW*, sendo as tarefas que serão revisadas, e *DONE*, sendo as histórias e tarefas finalizadas.

## 6 Desenvolvimento

### 6.1 Arquitetura do Sistema

Nesta imagem apresenta uma arquitetura simples de como o projeto se comportará.

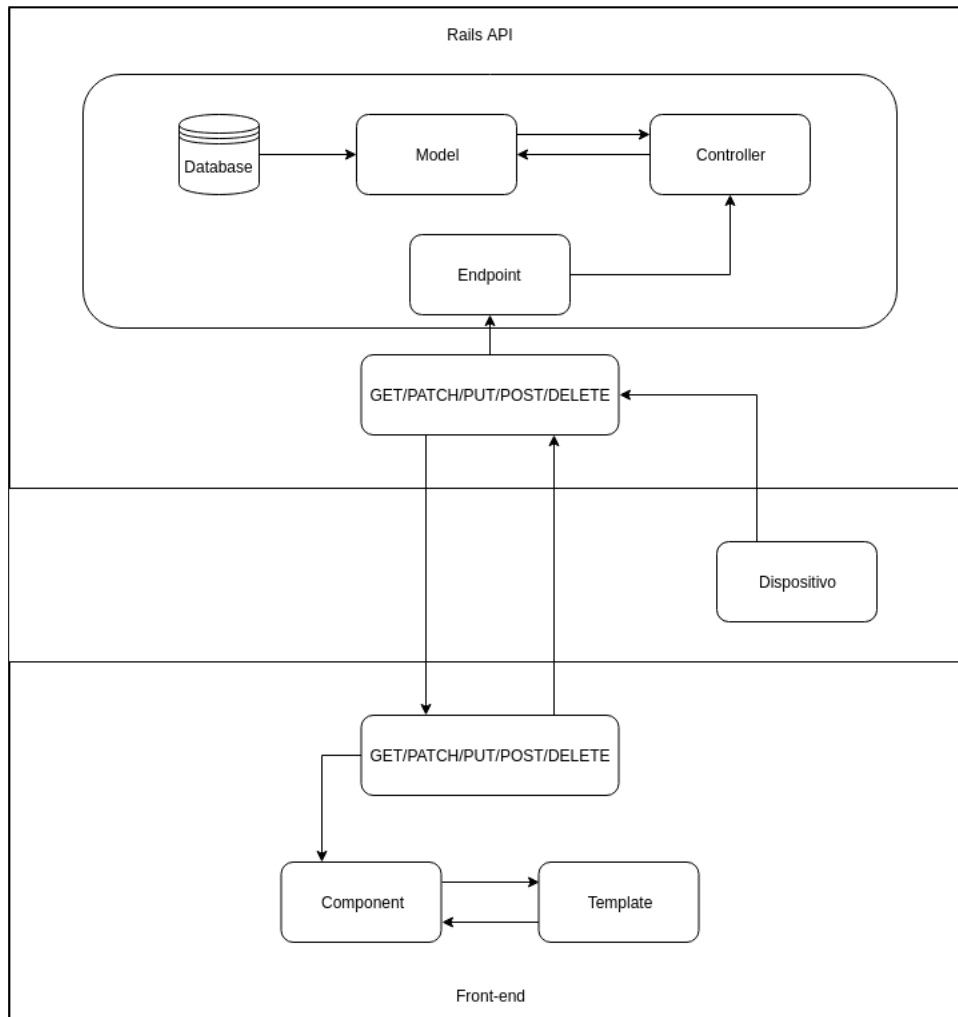


Figura 6 – Arquitetura do sistema

### 6.2 Protótipos de Tela

Para melhor demonstrar o sistema foram feitos protótipos de tela, demonstrando o *dashboard* da aplicação na figura 5, o formulário de cadastro do dispositivo na figura 6 e a tela de apresentação do dispositivo cadastrado na figura 7. Estes protótipos servirão de base para o desenvolvimento das telas do sistema.

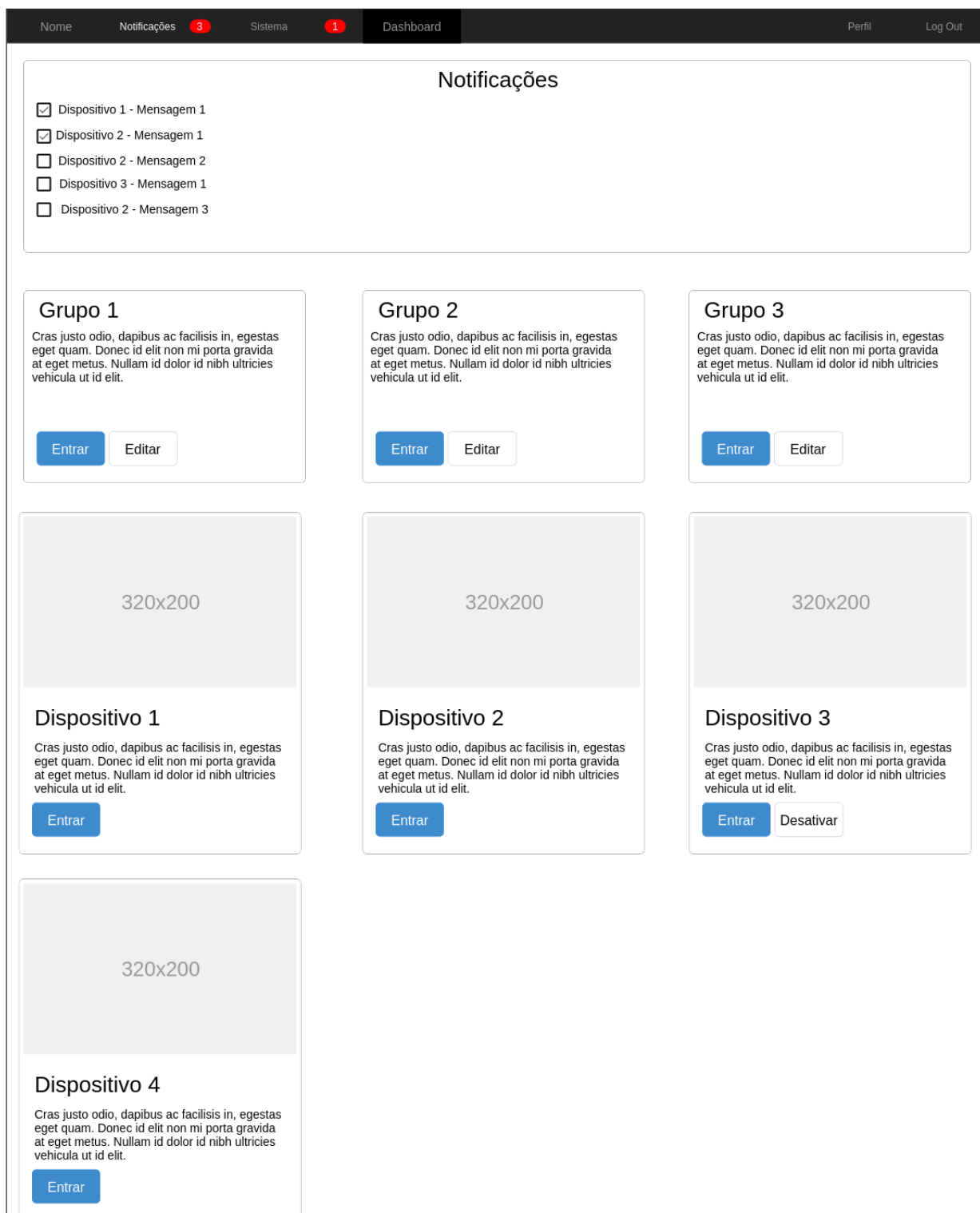


Figura 7 – Mockup - Dashboard aplicação

Nome   Dispositivos **15**   Sistema **5**   Statistics   Settings   Perfil   Log Out

Nome do dispositivo \*

IP \*

Descrição \*

320x200

Imagem do dispositivo

Selecionar imagem

Tipo do dispositivo: Sensor de incêndio   Grupo do dispositivo: Nenhum --

Adicionar   Cancelar

Figura 8 – Mockup - Formulário de cadastro de dispositivos

Nome   Notificações **3**   Sistema **1**   Dashboard   Perfil   Log Out

520x300

Dispositivo 1

Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.

Editar

**Notificações**

- Notificação 1 - Data: 25/11/2018 15:55
- Notificação 2 - Data: 24/11/2018 11:25
- Notificação 3 - Data: 10/08/2018 09:17
- Notificação 4 - Data: 11/05/2018 00:08
- Notificação 5 - Data: 10/12/2017 15:10

Ver todas

**Grupo 1**

Descrição: Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.

Entrar

Figura 9 – Mockup - Tela de apresentação do dispositivo

### 6.3 Diagrama de banco de dados

Para o projeto também foi desenvolvido um diagrama de banco de dados, para representação do sistema, como demonstrado na figura 8.

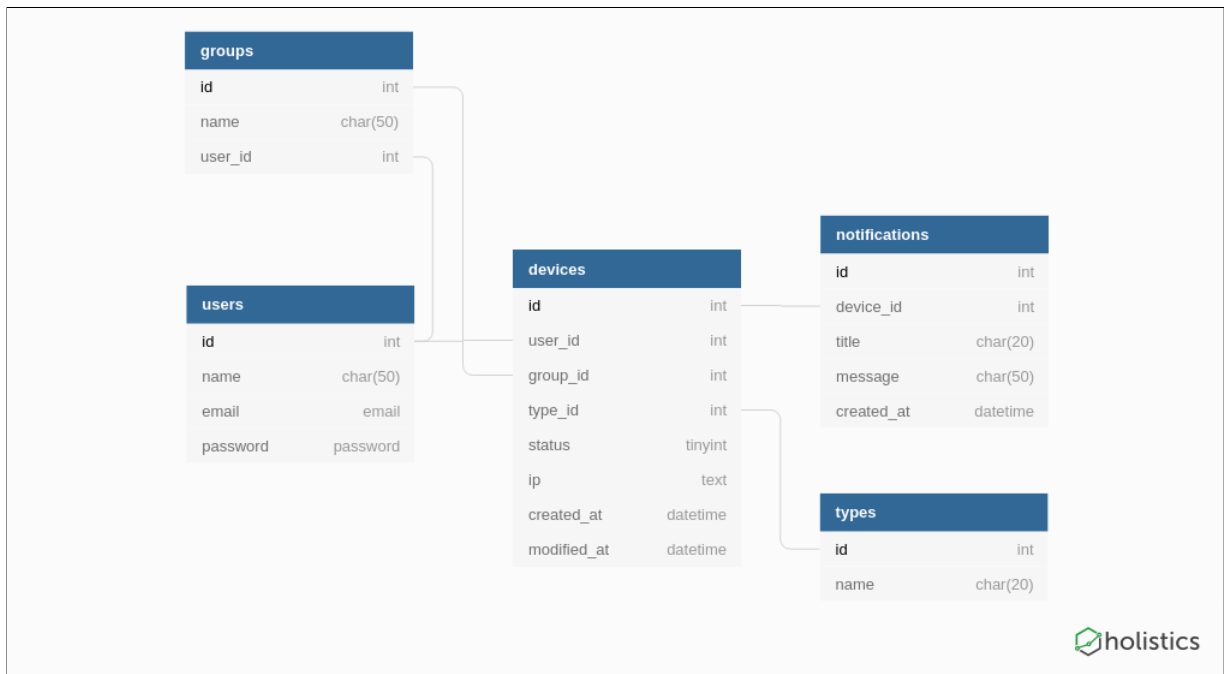


Figura 10 – Diagrama de Banco de dados

Fonte: [Holistics \(2018\)](#)

### 6.4 Histórias da Aplicação

Para o início do desenvolvimento foram escritas algumas histórias, que irão compor o *Product Backlog*, e poderão ser desenvolvidas durante o projeto. Porém essas histórias podem não ser desenvolvidas, assim como novas histórias podem surgir para o desenvolvimento.

- Como usuário eu quero me cadastrar e autenticar no sistema.
- Como usuário eu quero poder cadastrar meus dispositivos no sistema.
- Como usuário eu gostaria de poder criar grupos para organizar meus dispositivos.
- Como usuário eu gostaria de poder adicionar meus dispositivos aos meus grupos.
- Como usuário eu gostaria de poder ver as minhas notificações, grupos e dispositivos no *dashboard*.
- Como usuário eu gostaria de receber aviso das notificações na barra de navegação.
- Como usuário eu gostaria de consultar o estado de um dispositivo.

## 6.5 Cronograma

Quadro 1 – Cronograma de Atividades.

<b>Atividades</b>	<b>Ago</b>	<b>Set</b>	<b>Out</b>	<b>Nov</b>	<b>Dez</b>	<b>Jan</b>	<b>Fev</b>	<b>Mar</b>	<b>Abr</b>	<b>Mai</b>	<b>Jun</b>
1. Revisão dos apontamentos da banca		X									
2. Revisão bibliográfica		X	X								
3. Redação do projeto de TCC		X	X	X	X						
4. Defesa do projeto de TCC					F						
5. Desenvolvimento do ambiente simulado					F	F	F				
6. Desenvolvimento da aplicação Back-end							F	F	F		
7. Desenvolvimento da aplicação Front-end								F	F	F	
8. Escrita da Monografia de TCC								F	F	F	F
9. Elaboração da apresentação final											F
10. Defesa final do TCC											F

**X** - Finalizado **F** - Será realizado

## 7 RESULTADOS

Neste capítulo serão levantados os resultados já obtidos pelo projeto e os resultados esperados para o TCC2.

### 7.1 Resultados Parciais

- Entidade-Relacionamento
- Histórias da aplicação
- Arquitetura do sistema
- Quadro Kanban na plataforma Trello
- Histórias para desenvolvimento do sistema

### 7.2 Resultados Esperados

- Desenvolvimento da aplicação *Web back-end* em Ruby on Rails
- Desenvolvimento da aplicação *Web front-end* em Angular
- Desenvolvimento do ambiente simulado



## 8 CONSIDERAÇÕES FINAIS

Este projeto tem como finalidade auxiliar na segurança física de um ambiente, permitindo que as notificações de um dispositivo sejam alocadas na nuvem e possam ser acessadas de qualquer lugar pelo usuário. Além disso, é possível que o usuário acesse o histórico de notificação dos seus dispositivos, para que dessa forma ele possa saber quantas vezes o sensor foi ativado e poder identificar problemas no ambiente que está sendo vigiado pelos sensores.

Com o que foi posto, é também proposto que a aplicação seja desenvolvida em duas aplicações, separando o *front-end* do *back-end*, para que desta forma seja possível mudar as linguagens ou adicionar novas plataformas ao sistema, por exemplo adicionar um aplicativo *mobile* sem a necessidade de rescrever todo o código.

Para a comunicação entre os dispositivos será implantado o protocolo de comunicação CoAP, um protocolo desenvolvido para sistemas com baixo poder de processamento e bateria limitada, voltado também para *IoT*. Desta forma, possibilitando os dispositivos de trocar requisições com o *back-end*.

Para o desenvolvimento do projeto, as notificações serão vinculadas a modelos de dispositivos, que guardaram a referência de como encontrar o dispositivo físico. O usuário também poderá criar grupos, para organizar diversos dispositivos da maneira que lhe interesse.

## Referências

- ALMEIDA, H. **Tudo conectado – Internet das Coisas**. [S.l.], 2015. 58 p. Acesso em: 16 de Novembro de 2018. Citado na página 3.
- ANGULAR.IO. **Angular Guide**. 2018. Disponível em: <<https://angular.io>>. Acesso em: 27 de Novembro de 2018. Citado na página 7.
- ATLASSIAN. **Trello**. 2018. Disponível em: <<https://www.trello.com>>. Acesso em: 27 de Novembro de 2018. Citado na página 11.
- BIPM. **International vocabulary of metrology – Basic and general concepts and associated terms (VIM)**. 2008. Disponível em: <<https://jcgm.bipm.org/vim/en/3.8.html>>. Acesso em: 20 de novembro de 2018. Citado na página 7.
- BLUELUX. **Bluelux**. 2014. Disponível em: <<https://www.bluelux.com.br>>. Acesso em: 30 de Agosto de 2018. Citado na página 5.
- BORMANN, C. **RFC 7252 Constrained Application Protocol**. 2014. Disponível em: <<http://coap.technology/>>. Acesso em: 26 de Novembro de 2018. Citado na página 3.
- CARNEIRO, L. S. L. e Thiago de Aguiar Leal Domingues e V. **Develop an IoT application on Bluemix with Arduino and Rails**. [S.l.], 2014. Citado na página 4.
- CULTURAÁGIL. **Kanban do Início ao Fim**. 2014. Disponível em: <<https://www.culturaagil.com.br/kanban-do-inicio-ao-fim/>>. Acesso em: 26 de Novembro de 2018. Citado na página 10.
- HOLISTICS. **dbdiagram**. 2018. Disponível em: <<https://dbdiagram.io>>. Acesso em: 26 de Novembro de 2018. Citado na página 15.
- INMETRO. **Sistema Inmetro de Monitoramento de Acidentes de Consumo - Sinmac**. [S.l.], 2017. 60 p. Acesso em: 3 de novembro de 2017. Citado na página 1.
- KLIPP, P. **Getting Started with Kanban**. [S.l.], 2014. 36 p. Citado na página 10.
- RAILS, R. on. **Rails Guide**. 2007. Disponível em: <<https://guides.rubyonrails.org/getting-started.html>>. Citado na página 7.