

Proposta de um modelo de Rede Social das Coisas utilizando software livre

Trabalho de Conclusão do Curso de
Tecnologia em Sistemas para Internet

João Pedro Flach da Silva
Orientador: Andre Peres

¹Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Campus Porto Alegre
Av Cel Vicente, 281, Porto Alegre – RS – Brasil

joao.flachsilva@gmail.com, andre.peres@poa.ifrs.edu.br

Resumo. Neste trabalho é apresentada uma proposta de um sistema que permite a comunicação entre objetos reais no contexto da Internet das Coisas. A estrutura desenvolvida serve como repositório de dados produzidos por dispositivos eletrônicos dotados de sensores e/ou atuadores e como meio de disponibilização destes dados para outros dispositivos semelhantes. O trabalho possui como ponto de partida um sistema de rede social de código aberto chamado HumHub, o qual foi modificado para comportar este tipo de interação. Encontram-se neste artigo os detalhes técnicos sobre como a ferramenta HumHub foi adaptada para ter a capacidade de permitir que usuários a utilizem como uma Rede Social das Coisas. A validação do sistema desenvolvido foi feita através de uma simulação, com dados gerados por um serviço específico para este fim.

1. Introdução

A Internet das Coisas é o momento exato em que foram conectados à Internet mais coisas ou objetos do que pessoas [Evans 2011]. É uma extensão do que conhecemos hoje como Internet, alavancada pelos avanços tecnológicos em campos como microeletrônica, comunicação e sensoriamento. Tais avanços tornam possível configurar objetos comuns do dia a dia para que seja adicionada a capacidade de conexão com a Internet. A partir disto, viabilizam-se possibilidades como o controlamento remoto destes objetos e a utilização dos mesmos como serviços na Internet.

Neste cenário, a quantidade de dados disponíveis proporcionará um aumento sem precedentes do potencial de informação e conhecimento que podem ser gerados pela humanidade [Santos et al. 2016]. A temática deste trabalho será abordada ao redor do problema fundamental, que é fazer as Coisas se comunicarem através da Internet. Também é um propósito do trabalho promover uma aproximação entre o usuário comum de Internet e o conceito de Internet das Coisas. Primeiramente, serão abordadas duas definições contextuais importantes para o transcorrer do texto: o que é uma “Coisa” e o que significa “comunicar”.

É de conhecimento geral que a palavra “coisa” representa qualquer elemento que o conhecimento humano considera compreensível. Corriqueiramente utiliza-se esta palavra para referir-se a objetos materiais. Desta forma, dentro do contexto Internet das Coisas,

é possível considerar a aplicação do termo Coisa para se referir a algo grande e genérico, como uma cidade. Também pode servir para classificar elementos desta cidade, como prédios, veículos, ou até mesmo pessoas. De maneira ainda mais específica, Coisas podem se referir a micro partículas químicas, como bactérias, moléculas ou átomos.

Segundo [Gelernter 1992], um dia todas as Coisas estarão representadas virtualmente pelo que o autor chama de “mundo espelhado”. Cada “mundo espelhado” é um modelo de software de algum pedaço da realidade, um fragmento do mundo real acontecendo lá fora. Ainda segundo o autor, nestes modelos serão derramados oceanos de informação (através de um vasto labirinto de tubulações de software): tanta informação que o modelo poderá imitar cada movimento da realidade, momento a momento. As Coisas podem ser chamadas também de objetos inteligentes, que de acordo com [Kopetz 2011] são os blocos de construção da “Internet das Coisas”.

Sob a perspectiva deste trabalho, em relação à tecnologia da informação, a Coisa é um objeto físico equipado com sensores e/ou atuadores eletrônicos e um processador que utilizará um software para interpretar as entradas e saídas de dados. Ela pode servir para dois propósitos:

1. Captar dados do ambiente através de sensores eletrônicos;
2. Alterar o ambiente através do acionamento de atuadores.

Tanto a obtenção de dados quanto o envio de comandos deve ocorrer através de uma interface de comunicação entre Coisas.

Em um aspecto geral, a comunicação é o ato de transferir informação de um lugar para outro. Ela acontece quando uma entidade emissora passa uma mensagem para uma entidade receptora, utilizando algum tipo de linguagem conhecida por ambas. Assim, a segunda é capaz de utilizar as informações que recebeu para qualquer propósito que lhe seja conveniente. No contexto da “Internet das Coisas”, a comunicação consiste em fazer com que os dados coletados pelos sensores de uma Coisa sejam transmitidos pela Internet, atingindo a porta de entrada de dados de outras Coisas. Os dados recebidos podem servir de estímulo para que a Coisa receptora, por exemplo, acione seus atuadores para realizar alguma ação de qualquer natureza, que resultará na alteração do seu estado e/ou em interações diretas com o ambiente físico ao seu redor.

Com a perspectiva de endereçar a questão da comunicação entre as Coisas, discute-se em trabalhos acadêmicos como [Atzori et al. 2012] sobre a utilização de “Redes Sociais” em soluções para a “Internet das Coisas”, criando um novo conceito nomeado “Rede Social das Coisas”. Esta abordagem traria vantagens como a possibilidade de moldar a estrutura do sistema de forma a garantir a navegabilidade da rede, fazendo com que a descoberta de novos objetos e serviços seja realizada de maneira eficiente. Outro benefício seria a reutilização de modelos projetados para o estudo de “Redes Sociais”, endereçando questões relacionadas a uma rede extensa de objetos interconectados. Também um tópico de interesse seria o relacionamento entre os objetos inteligentes, de maneira semelhante às amizades em “Redes Sociais”. O estabelecimento de conexões deste tipo teria aplicabilidade em questões de segurança, utilizando por exemplo o nível de confiança entre dois objetos “amigos”.

Com base nas informações levantadas, o presente artigo explica a construção de uma solução tecnológica de software livre, criada para facilitar a integração dos compo-

mentes da Internet das Coisas. Para atingir este objetivo, foi desenvolvida uma plataforma nos moldes de uma Rede Social, na qual usuários conseguem cadastrar suas próprias Coisas, assim como procurar Coisas que pertencem a outros usuários. Ao registrar uma Coisa, o usuário compartilha com a comunidade os canais e detalhes técnicos para que os dados desta Coisa possam ser consumidos por outros sistemas, funcionando como um serviço. Através do sistema o usuário consegue visualizar e estabelecer conexões entre Coisas, configurando a maneira como dados provindos de uma Coisa irão alterar o estado de outra. Como objetivo secundário, este trabalho almeja difundir o conceito de Internet das Coisas na sociedade, aproveitando os recursos da Rede Social para apresentar Coisas em páginas de perfil, considerando que este tipo de apresentação é intuitiva para a percepção dos usuários comuns de Internet.

2. Estado-da-arte

Esta seção descreve algumas aplicações e trabalhos de pesquisa se assemelham ou se relacionam de alguma forma com o tema abordado neste trabalho.

2.1. Rede Social das Coisas

Na Internet das Coisas, os elementos que permeiam nosso cotidiano participarão ativamente de fóruns de troca de mensagens na Internet. Estas podem acontecer tanto entre Coisas, quanto entre Coisas e humanos. Este cenário demanda a existência de um sistema intermediário, que aglutine sob a mesma perspectiva informações sobre Coisas e humanos. Através de uma apresentação transparente e intuitiva das conexões entre essas entidades, alavanca-se maiores capacidades de configuração, manutenção e monitoramento.

Em [Atzori et al. 2012], é mencionado que existe um número relevante de pesquisas independentes investigando as potencialidades de integrar conceitos de redes sociais em soluções de Internet das Coisas, o que resultaria em um paradigma nomeado Rede Social das Coisas (traduzido do Inglês *Social Internet of Things (SIoT)*). Este traz o potencial de suportar novos tipos de aplicações e serviços para Internet das Coisas, de maneiras mais efetivas e eficientes.

As principais contribuições de [Atzori et al. 2012] são: identificar políticas apropriadas para o estabelecimento e gerenciamento de relacionamentos sociais entre Coisas, de maneira que a rede social resultante seja navegável; descrever uma possível arquitetura para a Internet das Coisas, que inclui as funcionalidades exigidas para integrar Coisas em uma rede social; analisar as características da estrutura de rede da *SIoT* por meio de simulações.

Ainda sobre [Atzori et al. 2012], o artigo aborda questões importantes como a escalabilidade e navegabilidade de uma *SIoT*. É descrito um modelo de criação de relacionamentos entre Coisas, utilizando o conceito de amizade em uma rede social. Este modelo tem o intuito de indexar a rede de Coisas, tornando-a navegável através de links que façam sentido, assim como escalável, no sentido de que existirão mais de trilhões de Coisas habitando este ambiente digital. Essa conexão acontece sem intervenção humana, sendo estabelecida pelo reconhecimento de um fator comum entre as Coisas. Por exemplo:

1. Relacionamento por origem: estabelecido entre Coisas que são de uma mesma leva de fabricação, oriundas de uma mesma origem;

2. Relacionamento por localização compartilhada: estabelecido entre Coisas utilizadas sempre no mesmo ambiente, como uma casa inteligente ou uma cidade inteligente;
3. Relacionamento por trabalho em equipe: estabelecido sempre que Coisas colaboram em favor de um serviço maior (por exemplo, pequenos sensores espalhados medindo a qualidade da água em um rio);
4. Relacionamento por propriedade: quando Coisas pertencem a um usuário (smartphones, smart tvs, consoles de video game, etc.);
5. Relacionamento social: quando Coisas entram em contato esporadicamente, devido a seus donos possuírem alguma relação social (amigo, colega, companheiro de viagem).

O trabalho de [Atzori et al. 2012] desempenha um papel fundamental no entendimento dos desafios que existem no campo da Internet das Coisas. Além disto, explica como podem ser endereçadas utilizando a abordagem de Rede Social questões importantes da Internet das Coisas, como navegabilidade de rede e escalabilidade.

O presente trabalho difere de [Atzori et al. 2012] ao focar em apresentar Coisas e pessoas sob uma mesma perspectiva em uma Rede Social. No sistema proposto, o usuário cadastra as Coisas e estas são vistas como uma página de perfil, semelhantemente à forma como humanos são apresentados. O usuário pode configurar conexões entre as Coisas, para que elas realizem comunicações. A manutenção destas conexões é feita através da página da Coisa.

2.2. Paraimpu

O Paraimpu é uma ferramenta que tem o propósito de prover ao usuário a capacidade de criar aplicações customizadas na Internet das Coisas. Para isso, ele deve primeiramente conectar suas Coisas ao sistema, utilizando os dois módulos disponibilizados: Sensores e Atuadores.

Segundo [Pintus et al. 2012], artigo escrito pelo fundador do Paraimpu, um Sensor é qualquer coisa capaz de produzir dados de um certo tipo (texto, numérico, JSON, XML entre outros); um Atuador, qualquer Coisa capaz de executar alguma ação através do consumo de dados a partir de um Sensor.

Na interface de usuário, ao acessar a tela de criação de Sensores, o usuário é apresentado a diversas categorias de fontes de dados, podendo criar, por exemplo: um Arduíno, um Twitter, um sensor genérico, entre outros. Na tela de Atuadores, as opções são para configurar as Coisas que estarão recebendo dados e executando alguma ação. Por exemplo, um Facebook para realizar postagens, um Arduíno, uma conta do Gmail, entre outros.

Após efetuar a criação de pelo menos um sensor e um atuador, é possível conectá-los, estabelecendo assim uma comunicação em tempo real entre os dois. Além disso, o usuário pode compartilhar os sensores e atuadores com outros usuários que são seus amigos no sistema, de forma que estes conseguem conectar suas Coisas comunitariamente. A comunicação é baseada na arquitetura *RESTful*, que utiliza protocolo HTTP para a troca de mensagens entre aplicações.

O Paraimpu oferece uma versão gratuita, que em relação a versão paga, tem um limite menor na quantidade de Coisas que podem ser criadas por um usuário. Em avaliação,

o sistema parece uma ideia que foi descontinuada, não atendendo de forma satisfatória o seu propósito.

2.3. IFTTT

O IFTTT [IFTTT 2018], uma abreviação em inglês para *If This Then That* - Se Isto, Então Aquilo - é um serviço web gratuito que possui funcionalidades para que um usuário possa configurar comunicações entre os seus aplicativos, proporcionando uma experiência mais integrada e fluída no dia a dia.

As conexões criadas entre os aplicativos dentro do IFTTT são chamadas de *applets*. Em um *applet* existe um aplicativo de origem, que serve como gatilho, e um aplicativo de destino, que será acionado quando o gatilho disparar. Em resumo, o usuário escolhe em que condição o aplicativo de origem fará o disparo do gatilho, para então o aplicativo de destino realizar alguma ação, também configurada pelo usuário. Os *applets* podem ser compartilhados com a comunidade de usuários do IFTTT.

O IFTTT é composto por módulos, que são integrados com APIs de uma vasta quantidade de aplicativos e produtos do mercado. Atualmente, esta plataforma possui integração com uma certa variedade de produtos da Internet das Coisas, como lavadoras de roupas, geladeiras, acessórios de casa inteligente e até mesmo eletrônicos DIY (do inglês *Do it Yourself* (faça você mesmo)).

O IFTTT apresenta uma solução robusta para Internet das Coisas, na qual o usuário é apresentado a uma quantidade considerável de possibilidades para integrar suas Coisas.

3. Conceitos abordados

Nesta seção há uma breve explicação para cada um dos conceitos que dão base para este trabalho.

3.1. Internet das Coisas

A Internet das Coisas foi definida como uma infraestrutura global para a sociedade da informação, habilitando a existência de avançados serviços, baseando-se em informações interoperáveis existentes e tecnologias de comunicação [ITU 2016].

É um novo paradigma tecnológico, que modifica o status quo no sentido de que, até então, a Internet era “das pessoas” e agora ela será de qualquer coisa. Isto significa que tudo que existe no mundo físico tem potencial para estar conectado de alguma forma à Internet, enviando a ela informações sobre o contexto em que está, assim como consumindo dados que nela se encontram.

Este cenário se tornou possível com o avanço tecnológico que vem ocorrendo nas últimas décadas. Frequentemente são anunciadas novidades que mostram que o espaço físico tem cada vez menos relevância quando se trata de computadores com alta capacidade de processamento. Hoje em dia um smartphone comum possui configurações que o tornam mais potentes que computadores pessoais (CPUs) do início do milênio.

Outro fato que contribui é a ampliação da infraestrutura da Internet no mundo, que possibilitou aumentar a velocidade da transmissão de dados. Também foram estabelecidos importantes protocolos como o Wi-Fi e o Bluetooth, que facilitam a conexão dos dispositivos e a troca de dados sem depender de um meio físico cabeado.

Aliando a utilização de pouco espaço físico e uma alta capacidade de processamento de dados, conectividade com a Internet e equipamentos eletrônicos com capacidade sensorial/atuadora, não há restrições para a criação de objetos com computadores acoplados. Pode-se pensar metaforicamente na adição de um modelo simplificado de cérebro às coisas, que lhes dá uma certa autonomia. Este computador apenas precisa ser capaz de se conectar à Internet para consumir e/ou produzir dados com algum objetivo. Cumprindo estes requisitos, pode-se classificar uma coisa como uma usuária da Internet. Esta é a premissa fundamental da Internet das Coisas.

3.2. Arduino

Arduino é uma plataforma *open source* baseada em hardware e software de fácil uso. Nasceu como uma ferramenta simples para prototipagem rápida, destinada a estudantes sem experiência em eletrônica e programação. Suas placas são capazes de ler entradas de dados - luzes em um sensor, um dedo em um botão ou uma mensagem no Twitter - e transformar em uma saída de dados - ativação de um motor, ligar um LED, publicar alguma coisa online.

Para o Arduino saber o que fazer, é necessário passar um conjunto de instruções para o microcontrolador acoplado. Isto é realizado através da utilização de uma linguagem de programação e um ambiente de desenvolvimento próprios do Arduino. Ao longo dos anos o Arduino tem sido o cérebro de milhares de projetos, de objetos cotidianos a complexos instrumentos científicos. Uma grande comunidade se reuniu em torno desta plataforma, o que acabou gerando uma vasta quantidade de conhecimento acessível, que pode servir de grande auxílio tanto para aprendizes quanto para especialistas.

Assim que atingiu uma comunidade mais larga, a placa Arduino começou a mudar para se adaptar às novas necessidades e desafios, diferenciando sua oferta de simples placas 8-bit para produtos de aplicações de internet das coisas, vestíveis, impressão 3D e ambientes incorporados. Todas as placas Arduino são completamente *open source*, empoderando usuários a construí-las independentemente e eventualmente adaptá-las às suas necessidades particulares. O software também é *open source* e está crescendo através das contribuições de usuários espalhados pelo mundo [Arduino 2016].

3.3. Yii2

O Yii é um framework de programação web genérico, o que significa que ele pode ser usado para o desenvolvimento de todo tipo de aplicações web baseadas usando PHP. Por causa de sua arquitetura baseada em componentes e suporte sofisticado a caching, ele é especialmente adequado para o desenvolvimento de aplicações de larga escala como portais, fóruns, sistemas de gerenciamento de conteúdo (CMS), projetos de e-commerce, Web services RESTful, e assim por diante [Yii2 2018].

3.3.1. Módulos

Módulos são unidades de software independentes que consistem em *models*, *views*, *controllers* e outros componentes de suporte. Os usuários finais podem acessar os *controllers* de um módulo quando ele é instalado no aplicativo. Por esses motivos, os módulos são geralmente vistos como mini-aplicativos. Os módulos são diferentes dos aplicativos, pois

os módulos não podem ser implantados sozinhos e devem residir em algum aplicativo [Yii2 2018].

3.4. HumHub

O *HumHub* é uma rede social de software livre e um *framework* que disponibiliza ferramentas para fazer com que a comunicação e colaboração seja fácil e bem sucedida [HumHub 2016]. Com este sistema, é possível criar uma rede social própria, através da instalação de uma instância em um servidor. Esta poderá ser customizada, pois não existem restrições para alterar o código fonte e adaptá-lo às necessidades encontradas. Foi desenvolvido utilizando a linguagem PHP e atualmente o framework Yii 2.0. O SGBD utilizado por padrão é o MySQL.

3.5. Tecnologias utilizadas

Abaixo a lista de tecnologias utilizadas para o desenvolvimento deste trabalho:

- PHP 5.6 e Yii2 Framework
- PHP Composer
- Bootstrap3 e Javascript
- WAMP Server
- Netbeans IDE 8.1 e Eclipse Oxygen IDE
- Git (para Windows) e GitHub
- Java 1.8 e Spring Framework (Boot e JPA)
- Google Chrome (Developer Tools)

4. Metodologia

Nesta seção explica-se como o trabalho proposto foi desenvolvido. No desenvolvimento o termo Coisa por vezes é substituído por *producer*, devido a uma maior aplicabilidade ao contexto em inglês dos códigos e documentações.

Para desenvolver o sistema proposto neste trabalho, implementou-se modificações no código da rede social *HumHub*, através da adição de um módulo *Yii2*, o qual recebeu o nome de *producer*. Este módulo adiciona ao *HumHub* a capacidade de cadastrar, descobrir e conectar *producers*. Desta forma, a estrutura funciona como um intermédio para comunicação entre Coisas. Através da página de perfil de um *producer*, o usuário tem acesso aos dados do *producer* e pode configurar conexões utilizando os serviços disponibilizados por HTTP.

Para realizar o processamento das conexões entre Coisas configuradas na Rede Social, foi criada uma aplicação de backend desacoplada do ambiente de infraestrutura HumHub. O único recurso compartilhado entre as duas aplicações é o banco de dados.

Os códigos estão disponíveis no GitHub:

- <https://github.com/JoaoFlach/humhub> - *branch* *iot_module* (HumHub com o módulo *producer*)
- <https://github.com/JoaoFlach/iot-simulator> - *branch* *master* (backend)

5. Desenvolvimento

Esta seção explica as etapas do desenvolvimento deste trabalho.

5.1. Pré-desenvolvimento

Esta seção explica o que foi necessário para preparar o ambiente de desenvolvimento.

5.1.1. Premissas

Abaixo são listadas algumas premissas técnicas que foram assumidas para construção das subsequentes seções do presente artigo:

- WAMP Server, Netbeans IDE, Git (para Windows), e PHP Composer (para Windows) instalados e configurados;
- Diretório de instalação do WAMP Server é “C:/wamp64”;
- Conta no GitHub para hospedar remotamente os repositórios Git.

5.1.2. Repositório do projeto

Tendo em vista o isolamento do repositório para o projeto, foi realizada uma operação de *fork* no repositório Git oficial do HumHub (<https://github.com/humhub/humhub>), criando um repositório separado hospedado no *GitHub*. A partir disto, copiou-se este repositório para a máquina de desenvolvimento, através da instrução “*git clone*”. Após executar este comando, os arquivos de projeto foram carregados para a pasta escolhida. Neste caso, foi a pasta “www”, na qual hospedam-se aplicações Web no Apache. Na figura 1 é possível visualizar como executar as instruções descritas acima utilizando o prompt de comando do Windows.



```

c:\ Prompt de Comando
Microsoft Windows [versão 10.0.17134.112]
(c) 2018 Microsoft Corporation. Todos os direitos reservados.

C:\Users\Flach>cd C:\wamp64\www

C:\wamp64\www>git clone https://github.com/JoaoFlach/humhub.git
```

Figura 1. Exemplo comando git clone

5.1.3. Dependências do projeto

Com os arquivos do projeto na máquina, foi necessário utilizar o PHP composer para que as bibliotecas que são dependências do HumHub fossem carregadas. Dentro da pasta raiz do HumHub, executou-se pela linha de comando as duas instruções abaixo:

- composer global require “fxp/composer-asset-plugin: 1.3”
- composer update

5.1.4. Configuração inicial do HumHub

Após finalizado o carregamento das dependências do projeto, foi possível acessar o endereço “http://localhost/humhub”. Em seu primeiro acesso, o HumHub forneceu telas para guiar a configuração inicial. O banco de dados foi criado automaticamente durante este processo. Ao final desta configuração, a aplicação de rede social estava pronta para ser utilizada.

5.1.5. URLs amigáveis

Para habilitar URLs amigáveis é preciso atualizar as configurações no arquivo “/protected/config/common.php”, de forma que o valor da propriedade *enablePrettyUrl* seja *true*, como na figura 2. Para esta configuração funcionar, o Apache precisa ter o módulo *mod-rewrite* ativado [Opitz 2006].

```
<?php
return [
    'components' => [
        'urlManager' => [
            'showScriptName' => false,
            'enablePrettyUrl' => true,
        ],
    ],
];
```

Figura 2. Exemplo habilitando *pretty URLs*

Na raiz do projeto há um arquivo chamado *.htaccess-dist* que precisa ser renomeado para *.htaccess*.

5.2. Adicionando o módulo Yii2 no HumHub

Para este trabalho foi criado um módulo chamado *producer*. A pasta raiz do módulo foi colocada abaixo da estrutura de arquivo “/protected/humhub/modules”. Na figura 3 é possível visualizar a estrutura de pastas e arquivos do módulo. No arquivo “Events.php” foi colocado um código que adiciona um item no menu superior horizontal. No arquivo “config.php” este código é associado com a construção do menu.

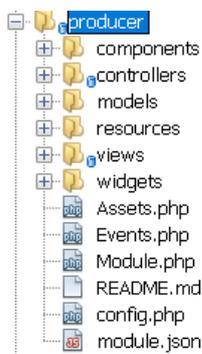


Figura 3. Estrutura de arquivos do módulo *producer*

5.3. Criação das tabelas de banco de dados

As tabelas adicionadas ao banco de dados são apresentadas no diagrama Entidade-Relacionamento (figura 4). As entidades são descritas na lista abaixo:

- **user**: armazena os dados dos usuários do *HumHub*. Nenhuma modificação foi feita em sua estrutura original.
- **producer**: criada para armazenar os dados de um *producer* (ou Coisa). Possui um relacionamento N para 1 com a tabela *user*, de maneira que um usuário seja o dono de um ou muitos *producers*.
- **producer_channel**: criada para armazenar um canal de comunicação de um *producer*. Basicamente, o endereço e o método de acesso de serviços HTTP. Possui um relacionamento N para 1 com a tabela *producer*. Sendo assim, um *producer* pode ter um ou muitos *producer_channel*.
- **producer_channel_property**: criada para armazenar as propriedades de um serviço HTTP de um *producer*. Possui um relacionamento N para 1 com a tabela *producer_channel*, de forma que um *producer_channel* pode ter uma ou muitas propriedades.
- **producer_connection**: criada para armazenar a configuração de conexões entre *producers*. Possui como chave estrangeira o identificador do serviço de origem (*origin_channel_id*), assim como o do serviço de destino (*post_channel_id*). Nas propriedades *when_property*, *condition_sign* e *condition_value* existem os dados que representam a expressão lógica que, quando verdadeira, permite que a comunicação entre origem e destino seja efetuada.

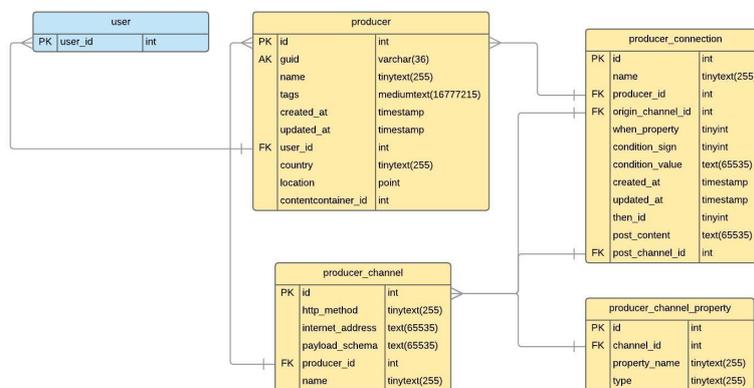


Figura 4. Diagrama Entidade-Relacionamento

5.4. Habilitar o módulo *producer* no HumHub

No HumHub, habilitar e desabilitar módulos é uma configuração administrativa, cabendo ao administrador do sistema fazer esse gerenciamento. Há uma interface disponível apenas para o usuário administrador, na qual pode-se habilitar módulos, como “Calendário” e “Aniversários”, que estendem as capacidades da Rede Social.

Para que as funcionalidades sendo desenvolvidas no módulo *producer* pudessem ser visualizadas e testadas, foi necessário habilitar o módulo *producer*. Na figura 5 é possível visualizar a tela onde esta configuração está disponível, mostrando o módulo *producer* habilitado.

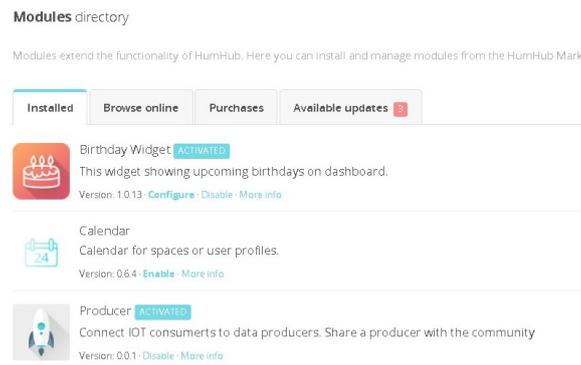


Figura 5. Habilitando o módulo *producer*

5.5. Interfaces de usuário

As interfaces de usuário do módulo *producer* foram construídas utilizando uma biblioteca auxiliar para criação de HTML do *Yii2 framework*, de maneira que os estilos existentes no HumHub foram reusados, mantendo uma aparência uniforme para a aplicação. Três telas foram criadas para as funcionalidades de cadastro, busca e apresentação dos dados de *producers*:

- Criar novo *producer*;
- Listar e buscar *producers* (figura 6);
- Perfil do *producer* (figura 6).

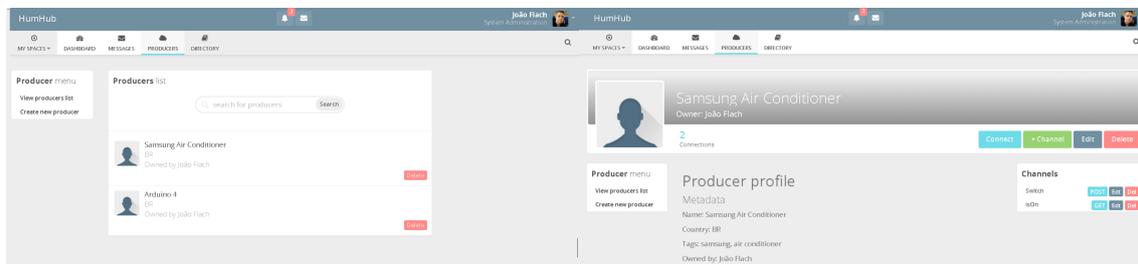


Figura 6. À esquerda, a *view* de listar e buscar *producers*. À direita, a *view* de perfil

A partir da *view* de perfil, é possível acessar as seguintes telas *modal* (pop-ups):

- Criar uma conexão para o *producer* (figura 8);
- Listar conexões do *producer*;
- Criar um novo canal para o *producer* (figura 7);
- Editar dados do *producer*;
- Deletar *producer*;
- Verificar resposta de um canal do *producer*.
- Editar um canal do *producer*;
- Deletar um canal do *producer*;

Na figura 9 é possível visualizar o diagrama de casos de uso do sistema.

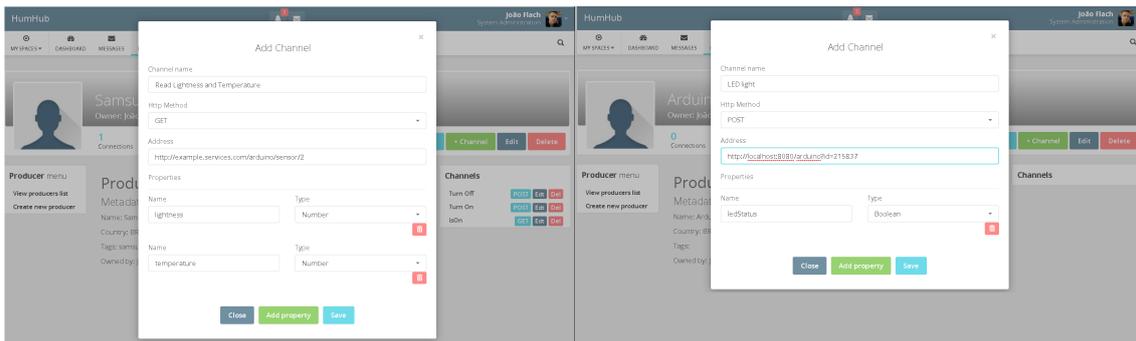


Figura 7. Criando canais

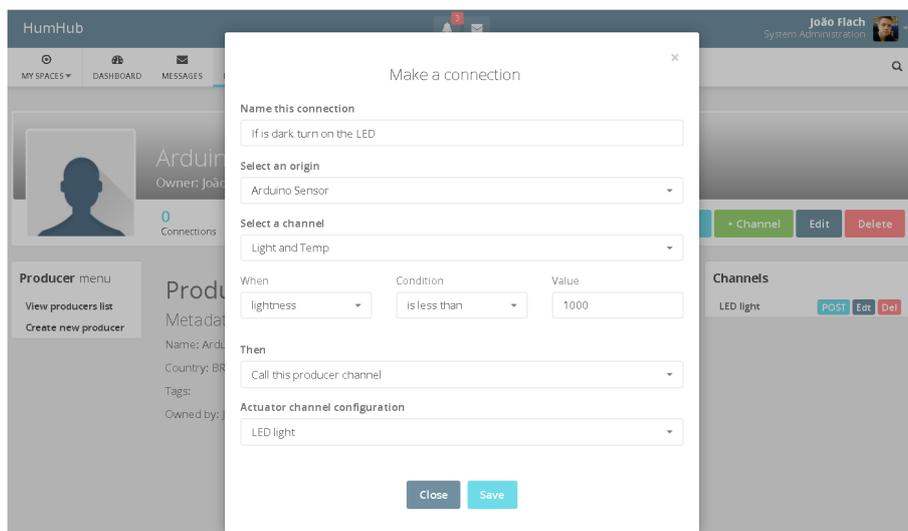


Figura 8. Criando uma conexão

5.6. Serviço backend para executar as conexões entre *producers*

Para realizar a comunicação definida nas conexões configuradas na Rede Social, foi criada uma aplicação de *backend* desacoplada do HumHub, utilizando a linguagem Java e o Spring Framework. Esta aplicação se conecta na mesma base de dados do HumHub, para consultar a tabela de conexões (*producer_connection*). Uma rotina agendada é executada com uma frequência de cinco (5) segundos. Durante a execução, uma consulta é realizada para recuperar os dados de todas as conexões existentes naquele momento. Caso existam conexões, os dados são colocados em uma lista para serem iterados. A iteração é feita de em serial, de maneira que as conexões são enfileiradas e processadas uma após a outra, sem processamento paralelo. Para cada iteração, as seguintes etapas são realizadas:

1. Solicitação do serviço HTTP de leitura do canal do *producer* origem;
2. Mapeamento das propriedades recebidas conforme o canal do *producer* foi configurado na Rede Social;
3. Avaliação da expressão lógica da conexão;
4. Caso a expressão lógica seja verdadeira, solicita o serviço de escrita no canal do *producer* destino, conforme configurado na conexão;
5. Caso a expressão lógica seja falsa, ignora o processamento da conexão;
6. Após processar a última conexão da lista, termina a rotina.

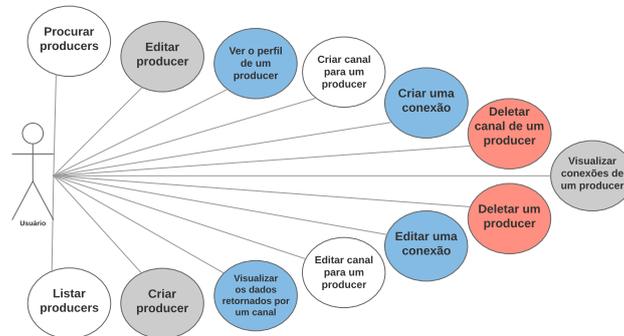


Figura 9. Diagrama de casos de uso

A figura 10 mostra um diagrama da arquitetura de alto nível, onde a aplicação backend aparece sob a alcunha de “Processador de Conexões IoT”.

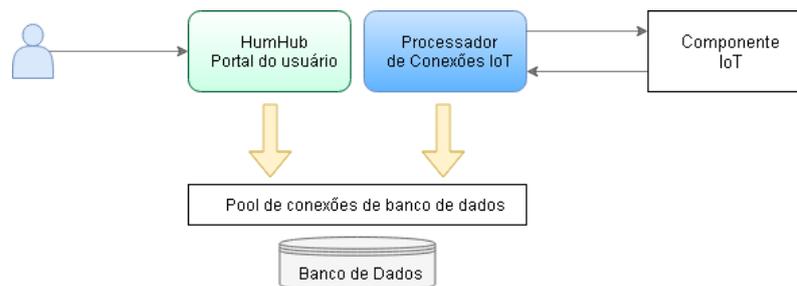


Figura 10. Diagrama da arquitetura de alto nível

6. Validação

Para fins de validação do trabalho, foi executado um cenário de teste com dados simulados em uma aplicação Java específica para este fim. Foram definidas duas classes, cada uma representando um tipo de placa Arduino:

- Arduino Sensor: com uma propriedade para guardar o dado simulado de um sensor de luminosidade.
- Arduino Atuador: com uma propriedade para guardar o valor *boolean* do estado simulado de uma lâmpada LED

Foram criadas interfaces HTTP para recuperar o valor do sensor de luminosidade e para mudar o estado da lâmpada LED. Com isso, criou-se no HumHub um *producer* para cada um destes objetos simulados. Foi adicionado ao perfil do Arduino Sensor o canal de comunicação de leitura do sensor de luminosidade, assim como no Arduino Atuador foi colocado o canal de comunicação de escrita para alterar a lâmpada LED. Após essa configuração, foi criada uma conexão entre os *producers*, na qual quando o valor de luminosidade recebido for um número negativo, é acionado o serviço HTTP para acender a lâmpada LED.

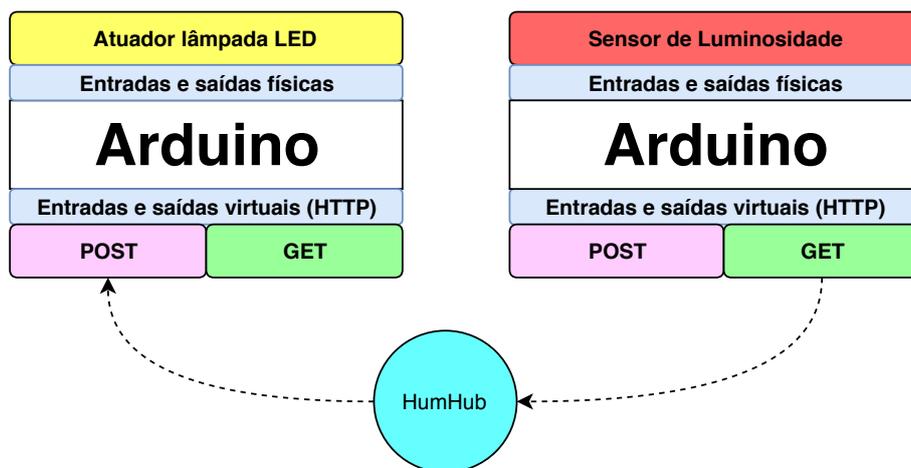


Figura 11. Ilustrando o cenário de validação

Para observar os resultados da validação realizada, utilizou-se os *logs* da aplicação backend. Ao final do teste, foi possível constatar que ele atingiu o resultado esperado, visto que o valor da lâmpada LED simulada se tornava *true* quando o valor do sensor de luminosidade simulado era menor que zero (0).

7. Considerações finais

O trabalho proposto considera o estado atual do desenvolvimento tecnológico, responsável por dispositivos eletrônicos menores, mais baratos e disponíveis, o qual produz um novo foco na fabricação de coisas conectadas com pessoas e entre si. Neste sentido, a possibilidade de inclusão de conexão à rede das coisas existentes e no desenvolvimento de novos produtos traz consigo a necessidade de fornecer um mecanismo que sirva de repositório de dados, que serão gerados por estes novos objetos. Desta forma, este repositório servirá como fórum de comunicação e interação.

Visando contribuir no desenvolvimento de soluções para este tema, a estrutura proposta por este trabalho utiliza um sistema de rede social com código aberto, alterando este sistema para que comporte a troca de dados descrita. Objetiva-se com este estudo fornecer um modelo inicial de troca de dados entre coisas utilizando a rede social *HumHub* para armazenamento e consumo dos mesmos por objetos equipados com controladores Arduino.

O sistema foi validado através de métodos de simulação, considerando que serviços disponíveis por HTTP podem ser abstração de controladores Arduino.

8. Trabalhos futuros

Existem oportunidades para trabalhos futuros, como os listados abaixo:

- Automatização de postagens na Rede Social baseada em dados provindos de um grupo de Coisas
- Implementação de relacionamentos entre as entidades de uma Rede Social das Coisas, com foco na melhora da navegabilidade da rede (big data, machine learning)

- Transformação de uma Rede Social das Coisas em um sistema altamente escalável, alterando a estratégia de processamentos de conexões para uma abordagem distribuída (paralelismo, threads, clusters)
- Desenvolvimento e validação da experiência de usuário de uma Rede Social das Coisas
- Desenvolvimento de uma ferramenta de busca avançada para uma Rede Social das Coisas

Referências

- Arduino (2016). Arduino - introduction. Disponível em: <https://www.arduino.cc/en/guide/introduction>. Acessado em Julho de 2018.
- Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012). The social internet of things (siot)—when social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks*, 56(16):3594–3608.
- Evans, D. (2011). A internet das coisas. *San José: Cisco IBSG*.
- Gelernter, D. (1992). *Mirror worlds: Or the day software puts the universe in a shoebox... How it will happen and what it will mean*. Oxford Paperbacks.
- HumHub (2016). Humhub - the flexible open source social network kit for collaboration. Disponível em: <https://www.humhub.org/en>. Acessado em: Julho de 2018.
- IFTTT (2018). Ifttt - a world that works for you. Disponível em: <https://ifttt.com/>. Acessado em Julho de 2018.
- ITU (2016). Internet of things global standards initiative. Disponível em: <http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx>. Acessado em: Junho de 2016.
- Kopetz, H. (2011). Internet of things. In *Real-time systems*, pages 307–323. Springer.
- Opitz, P. (2006). Clean urls for a better search engine ranking. Disponível em: <http://www.contentwithstyle.co.uk/content/clean-urls-for-a-better-search-engine-ranking/>. Acessado em Julho de 2018.
- Pintus, A., Carboni, D., and Piras, A. (2012). Paraimpu: a platform for a social web of things. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 401–404. ACM.
- Santos, B. P., Silva, L., Celes, C., Borges, J. B., Neto, B. S. P., Vieira, M. A. M., Vieira, L. F. M., Goussevskaia, O. N., and Loureiro, A. (2016). Internet das coisas: da teoria a prática. *Minicursos SBRC-Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.
- Yii2 (2018). The definitive guide to yii 2.0. Disponível em: <https://www.yiiframework.com/doc/guide/2.0/en/structure-modules>. Acessado em Julho de 2018.