

**GUILHERME SANCHES PEREIRA
JÉSSICA ADRIELE DO NASCIMENTO**

**GERENCIADOR DE CONSUMO DE ENERGIA ELÉTRICA
RESIDENCIAL**

**UNIVERSIDADE DO VALE DO SAPUCAÍ
POUSO ALEGRE
2016**

**GUILHERME SANCHES PEREIRA
JÉSSICA ADRIELE DO NASCIMENTO**

**GERENCIADOR DE CONSUMO DE ENERGIA ELÉTRICA
RESIDENCIAL**

Pré-projeto de conclusão de curso, solicitado pelos professores (as) Joelma Pereira de Faria e Carolina Padilha Fedatto, da disciplina de Trabalho de Conclusão de Curso – 1, do curso de Sistemas de Informação da Universidade do Vale do Sapucaí.

**UNIVERSIDADE DO VALE DO SAPUCAÍ
POUSO ALEGRE
2016**

LISTA DE FIGURAS

Figura 1 - Modelo plataforma Arduino UNO REV 3.....	10
Figura 2 - <i>Raspberry Pi</i> 2.....	13
Figura 3 - Modelo de Sensor de Corrente TC SCT 013-000.....	14

LISTA DE ABREVIATURAS E SIGLAS

AC	Corrente Alternada
ANSI	<i>American National Standards Institute</i>
API	<i>Application Programming Interface</i>
CSS	<i>Cascading Style Sheet</i>
HTML	<i>HyperText Markup Language</i>
IDE	<i>Integrated Development Environment</i>
IOT	<i>Internet of Things</i>
LCD	<i>Liquid Crystal Display</i>
REST	<i>Representational State Transfer</i>
TC	Transformador de Corrente
URI	<i>Uniform Resource Identifier</i>

SUMÁRIO

INTRODUÇÃO.....	5
2 JUSTIFICATIVA	6
3 OBJETIVOS.....	7
3.2 Objetivo geral	7
3.3 Objetivos específicos.....	7
4 QUADRO TEÓRICO.....	8
4.1 Internet das Coisas.....	8
4.2 Linguagem C	9
4.3 Arduino	9
4.4 <i>Raspberry Pi</i>	12
4.5 Sensor de corrente TC	13
4.6 <i>HyperText Markup Language</i> (HTML).....	14
4.7 <i>Cascading Style Sheet</i> (CSS).....	14
4.8 <i>JavaScript</i>	15
4.9 <i>AngularJS</i>	15
4.10 MYSQL	16
4.11 NodeJS.....	16
4.12 <i>JavaScript Object Notation</i> (JSON)	16
4.13 RESTFUL.....	17
4.14 <i>JQuery</i>	18
4.15 <i>Materialize</i>	18
5 REFERÊNCIAS	19

INTRODUÇÃO

O gasto com energia elétrica residencial está no cotidiano de uma grande massa populacional, que não possui a sua disposição qualquer mecanismo que lhe possibilite a visualização de seus gastos parciais com auxílio de ferramentas intuitivas, como gráficos e/ou imagens. Este fato, juntamente com o conceito atual de consumismo exacerbado, contribui para que haja um aumento da fatura energética ao final do mês, gerando insatisfação na maioria das vezes com algum órgão público ou até mesmo com a companhia distribuidora de energia local.

Visando este fator de insatisfação presente em diversas casas ao redor de nossa região no que tange aos altos valores nas faturas de energia elétrica, propomos a construção de uma arquitetura de placas eletrônicas com sensores, que realizem a medição da corrente elétrica em cada cômodo residencial e enviem essas informações para um banco de dados nas nuvens. Esses dados serão coletados, analisados e disponibilizados pelo sistema de gerenciamento *online*, que poderá ser acessado pelos moradores da residência de qualquer dispositivo que possua conexão com a internet, a fim de verificarem seus gastos parciais através de gráficos informativos detalhados de cada sensor/cômodo e mensagens de alerta que irão expor aos usuários, fatos que possam estar ocorrendo em sua residência, como por exemplo, um alto gasto energético em algum cômodo por um curto período de tempo.

Até a data atual, alguns projetos científicos foram elaborados com objetivo de realizar a mesma medição que sugerimos, porém, as exibições de resultados são mais simples: sendo por envio de mensagens de texto, relatórios em lista corrida, exibição em um *display* LCD. Portanto, o presente projeto tem como objetivo a disponibilização de uma interface gráfica e *online* que possa ser conectada à arquitetura de sensores já citada acima e a mesma ser verificada de qualquer computador com acesso à internet, suprimindo assim o baixo uso de recursos gráficos e intuitivos para usuários mais leigos que possuem esta carência de informação e contribuir para a comunidade tecnológica empregando recursos de *IoT*¹, disponibilizando-os em forma de trabalho científico.

¹ IOT – Abreviação para Internet of Things (Internet das Coisas)

2 JUSTIFICATIVA

Analisando nossa comunidade local, seja através de uma conversa informal com vizinhos, ou até mesmo ouvindo histórias enquanto estamos usando um transporte coletivo, escutamos em diversos casos, reclamações e insatisfações da população a respeito de valores de contas referentes ao saneamento básico: água, luz e gás. Recentemente, temos visto nas mídias sociais que devido à uma crise financeira a qual o país está passando, os impostos sobre alguns serviços básicos como água e luz estão tendo uma ascensão e quem paga por isto é o consumidor final.

Todo este contexto econômico, político e social faz com que a população, insatisfeita com estas alterações, modifique seus hábitos a fim de reduzirem gastos nestes serviços e, conseqüentemente, manterem seu padrão orçamentário ao final do mês. Neste ponto de vista atual da comunidade, o tema proposto surge como um intensificador no controle de gastos, especificamente nos gastos com energia elétrica, pois irá fornecer diversos gráficos de uso específicos por cômodo da residência além de expor ao usuário, mensagens, alertas e informações sobre comparações do mês atual com outros dados de meses já existentes no banco de dados.

O projeto também contribui para a comunidade tecnológica fornecendo mais um material científico que poderá servir de base para diversos outros projetos semelhantes e trazer cada vez mais para a atualidade o conceito de *IoT*, que segundo Gustavo Travassos (2015), “[...] apresenta, em nível global, uma enorme área de oportunidades, mesmo estando no início de sua maturidade em nosso país. ”

Outro ponto a ser citado refere-se à arquitetura utilizada no projeto a qual irá contar com total desacoplamento das três áreas que o projeto irá contemplar: *hardware*, serviço e aplicação web. Este fato traz para a atualidade o conceito de independência entre as camadas que faz com que qualquer uma delas possa ser completamente alterada sem necessitar qualquer alteração nas demais, resultando em uma alta manutenibilidade do sistema.

3 OBJETIVOS

Neste capítulo serão descritos os objetivos estabelecidos pelos integrantes da equipe, a fim de obterem o resultado proposto por este projeto.

3.2 Objetivo geral

- Desenvolver uma aplicação *web* capaz de gerenciar os gastos com energia elétrica residencial.

3.3 Objetivos específicos

Para alcançar o objetivo geral, será necessário cumprir as etapas:

- Coletar e medir a energia elétrica dos cômodos de uma residência através de placas eletrônicas com sensores;
- Armazenar os dados coletados através da *cloud computing*²;
- Criar um ambiente *online* para acesso e verificação das informações referentes ao consumo de energia elétrica.

² CLOUD COMPUTING – Também conhecida como, computação nas nuvens, fala-se na possibilidade de acessar arquivos e executar diferentes tarefas pela internet. Quer dizer, você não precisa instalar aplicativos no seu computador para tudo, pois pode acessar diferentes serviços online para fazer o que precisa, já que os dados não se encontram em um computador específico, mas sim em uma rede.

4 QUADRO TEÓRICO

Para desenvolvimento do projeto, será empregado o uso de diversas tecnologias, ferramentas e conceitos que serão descritos neste capítulo.

4.1 Internet das Coisas

O termo *Internet of Things* (Internet das Coisas) foi dito pela primeira vez, em 1999, por Kevin Ashton, cofundador do Auto-ID Center do *Massachusetts Institute of Technology* (MIT). Em artigo, Ashton (2009) alegou que a ideia original do termo Internet das Coisas previa a conexão entre todos os objetos físicos à internet, com capacidade de absorver informações por meio de identificação por radiofrequência e tecnologias de sensoriamento - as quais os permitiriam observar, identificar e compreender o mundo independentemente das pessoas e suas limitações de tempo, atenção e precisão.

A internet, hoje conta com quase três bilhões de usuários conectados segundo “*Global Inernet Report*” (KENDE, 2014). E segundo previsões de (STAMFORD, 2013), em 2020, o número de dispositivos conectados e interligados será de 26 bilhões. Algumas previsões mais otimistas, segundo a CISCO, prevê 50 bilhões de objetos conectados no mesmo período, gerando uma movimentação de mercado de US\$ 14,4 trilhões até 2022 (EVANS, 2011).

Portanto, segundo (LACERDA e LIMA-MARQUES, 2015, p. 161):

"As inovações que surgem no âmbito da Internet das Coisas ampliam o potencial humano em diversas áreas - tais como planejamento urbano (cidades, edifícios e trânsito inteligentes), meio ambiente (energia, água), indústria, comércio, turismo, educação, saúde, trabalho, segurança, programas sociais, governo - com soluções capazes de promover desenvolvimento econômico, sustentabilidade e qualidade de vida."

Tais fatos trazem para sociedade atual, uma demanda de pessoas, órgãos e empresas privadas, capazes de desenvolverem e programarem recursos de internet das coisas como forma de gerarem lucro financeiro e se inserirem em um mercado com ampla expansão. Este conceito de uma nova geração de conectividade, alinhado ao uso de novas tecnologias e ferramentas disponíveis no mercado, justificado pelo atual estado de crescimento tecnológico torna-se viável e eficaz o uso de recursos de *IoT* neste projeto.

4.2 Linguagem C

A linguagem C foi criada em 1972 em um laboratório chamado *Bell Telephone Laboratories*, por Dennis Ritchie, para que fosse permitido a escrita de um sistema operacional, o UNIX, que utilizasse os benefícios de uma linguagem de alto nível, deixando de lado o uso de linguagens de baixo nível que detinham de uma maior complexidade para seu uso (DAMAS, 2007).

Inicialmente, a linguagem servia para desenvolvimento de programas de sistemas, que são, segundo Schildt (1986, p.2), “[...] uma classe de programa que, ou são parte, ou operam em conjunto com o sistema operacional do computador [...]”.

Atualmente, após mais de quatro décadas de sua criação, a linguagem se apresenta ao mercado também em dispositivos eletrônicos de pequeno porte como micro controladores pelo fato de ser uma linguagem de fácil portabilidade. Tal expansão da linguagem pode ser observada em Klotz, (2010, p.3) que diz que “uma das melhores características de C é que não está ligada a qualquer equipamento ou sistema em particular. Isto torna mais fácil para um usuário para escrever programas que serão executados sem quaisquer alterações em praticamente todas as máquinas”³.

Devido a todos estes conceitos teóricos e históricos no que tange a abrangência da linguagem C em dispositivos embarcados, torna-se viável o uso da mesma no projeto, atuando na plataforma de desenvolvimento embarcado, o Arduíno, fazendo toda a leitura da corrente elétrica e enviando os dados para uma central receptora, que será outra plataforma para desenvolvimento, a *Raspberry*.

4.3 Arduino

Arduino é uma plataforma de desenvolvimento e prototipagem *open-source* que, segundo Evans, Noble e Hochenbaum (2013, p.25) “a Arduino teve início no *Interaction Design Institute* de Ivrea, na Itália, em 2005. ”

³ Original “One of the best features of C is that it is not tied to any particular hardware or system. This makes it easy for a user to write programs that will run without any changes on practically all machines”. Tradução feita pelos próprios autores.

Esta plataforma teve sua criação, destinada a ajudar estudantes de uma universidade italiana que tinha como premissa: “[...] o preço almejado não poderia ser mais do que um estudante gastaria se saísse para comer uma pizza”, (EVANS, et al., 2013, p.25).

Após sua criação, foram vendidas rapidamente as unidades fabricadas e a mesma começou a ser amplamente divulgada nas universidades da Itália e mais tarde pelo mundo todo.

Esta plataforma trabalha no conceito de entrada de dados, através de portas eletrônicas e/ou digitais, realiza-se um processamento dos dados e gera-se uma saída, sendo este procedimento executado em *loops* predefinidos pela linguagem de programação.

A linguagem utilizada nesta plataforma é a linguagem C, que como citada anteriormente, tornou-se uma linguagem adaptável a qualquer dispositivo que siga o padrão ANSI definido para a linguagem.

Segundo o site oficial da plataforma, no que diz respeito a evolução de sua criação:

Ao longo dos anos, Arduino tem sido o cérebro de milhares de projetos, desde objetos do cotidiano para instrumentos científicos complexos. A comunidade mundial de tomadores - estudantes, amadores, artistas, programadores e profissionais - reuniu em torno desta plataforma open-source, suas contribuições acrescentaram-se a uma quantidade incrível de conhecimento acessível que pode ser de grande ajuda para novatos e especialistas similares (ARDUINO, 2016, p.1). (Disponível em: <www.arduino.cc/en/Guide/Introduction>)

O *hardware* que compõe a plataforma Arduino é variado conforme os diversos modelos que foram lançados desde seu lançamento em 2005. Para este projeto iremos analisar o modelo UNO REV 3:

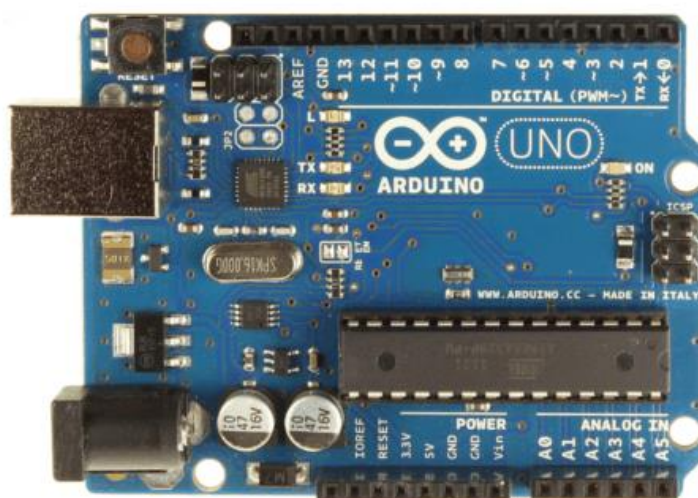


Figura 1 - Modelo plataforma Arduino UNO REV 3. **Fonte:** http://www.embarcados.com.br/wp-content/uploads/2013/11/imagem_01.png (acesso em 23 de fevereiro de 2016)

Como visto na Figura 1, a Arduino é composta por:

- 14 pinos de entrada e saída digital (pinos 0-13):

Esses pinos podem ser utilizados como entradas ou saídas digitais de acordo com a necessidade do projeto.

- 6 pinos de entradas analógicas (pinos A0 - A5):

Esses pinos são dedicados a receber valores analógicos, por exemplo, a tensão de um sensor. O valor a ser lido deve estar na faixa de 0 a 5 V onde serão convertidos para valores entre 0 e 1023.

- 6 pinos de saídas analógicas (pinos 3, 5, 6, 9, 10 e 11):

São pinos digitais que podem ser programados para ser utilizados como saídas analógicas.

A alimentação da placa pode ser feita a partir da porta USB do computador ou através de um adaptador AC. Para o adaptador AC recomenda-se uma tensão de 9 a 12 volts.

Juntamente na placa existe um processador da família Atmel que muda em cada modelo de Arduino.

Para se desenvolver nesta plataforma, é necessário o uso de um *software* que faça *upload* do código programado em linguagem C para dentro do micro controlador embutido na placa e que o mesmo seja executado quando a Arduino for conectada à rede elétrica. Segundo o site da equipe Embarcados (2013) para se programar na Arduino é necessário a utilização de uma IDE que permite a criação de um código para a placa e após a escrita do código deve-se clicar sobre o botão de *upload* da IDE que então irá traduzir este código feito em linguagem C para uma outra linguagem que possa ser compreendida pelo micro controlador.

Neste projeto a Arduino será responsável pela conectividade com os sensores que farão a leitura da energia elétrica e disponibilizará estes dados para a central *Raspberry* conectada à nuvem. Sua facilidade de aprendizado e compra por um baixo custo foram fatores relevantes na escolha desta plataforma.

4.4 Raspberry Pi

“A *Raspberry Pi* é uma máquina completa, com considerável poder de processamento, em uma placa de circuito impresso menor do que um cartão de crédito” (UPTON; HALFACREE, 2013, p. 26).

Foi desenvolvida, em 2006, por Eben Upton, Rob Mullins, Jack Lang e Alan Mycroft, com base na Universidade de Laboratório de Informática de Cambridge e na empresa *Atmel ATmega*, que desenvolve placas com processadores ARM. Eben Upton e sua equipe, em 2009, oficialmente estabeleceram a *Raspberry Pi Foundation*, fundação de caridade educacional, com base no Reino Unido, que tem como meta, permitir que as pessoas de todas as idades possam aprender a programar e entender como funcionam os computadores. (RASPBERRY PI, 2014).

O coração do dispositivo *Raspberry Pi* é o processador multimídia *Broadcom BCM2835*, que tem a maioria dos componentes do sistema montado em um único e é baseado na arquitetura de conjunto de instruções ARM, desenvolvida pela *Acorn Computers*, no final dos anos 80, sendo ele, capaz de operar apenas com alimentação de energia de 1A e 5V, fornecida pela sua porta micro-USB. O baixo consumo do chip é traduzido, diretamente, em pouco desperdício de energia, mesmo durante tarefas complexas de processamento. Outra diferença importante entre o *Raspberry Pi* e seu desktop, sem falarmos no tamanho e no preço, é o sistema operacional – Linux, que possibilita fazer download de todo o código-fonte referente ao sistema operacional e realizar quaisquer alterações necessárias. (UPTON; HALFACREE, 2013).

Segundo Upton (2013), existem dois modelos da placa *Raspberry Pi*: Modelo A e Modelo B. A diferença entre eles é que o modelo B, possui uma placa Ethernet e duas portas USBs, enquanto a modelo A, contém apenas uma porta USB e nenhuma Ethernet.

Para este projeto será utilizada o modelo *Raspberry Pi 2 model B+*, devido sua melhor configuração de *hardware* e aplicabilidade do projeto.

A Figura 2 apresenta o microprocessador *Raspberry Pi 2 Model B+*:

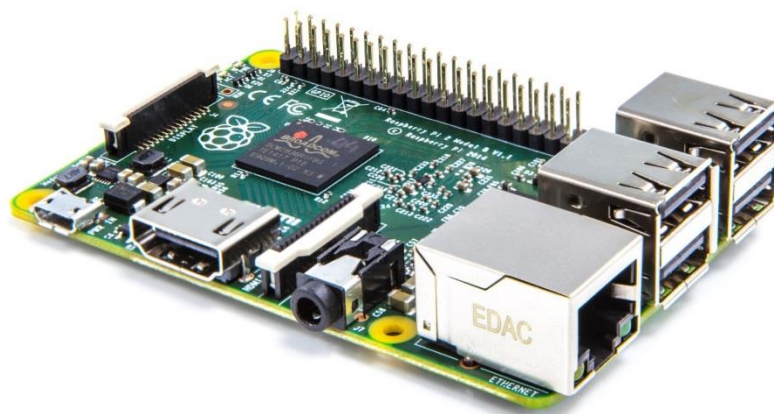


Figura 2 - Raspberry Pi 2 Fonte: https://www.raspberrypi.org/wp-content/uploads/2015/01/Pi2ModB1GB_-comp.jpeg (acesso em 20 de março de 2016).

Neste projeto, a *Raspberry Pi* receberá os dados coletados pela Arduino, apresentada na subseção 4.3 por meio dos sensores eletrônicos de corrente TC, que serão apresentados na subseção 4.5 e enviará as informações para o banco de dados online.

4.5 Sensor de corrente TC

Os sensores de correntes TC são dispositivos eletrônicos desenvolvidos para serem aplicados em diversos circuitos elétricos, através de variadas plataformas existentes de prototipagem (Arduino PIC, *Raspberry*) para mensurarem a corrente elétrica de algum dispositivo. Ele será conectado diretamente à Arduino e fará a mensuração dos valores de corrente de cada cômodo/disjuntor e enviará os dados para a plataforma de desenvolvimento.

Existem diversos modelos no mercado com diversas potências máximas de mensuração. Para o presente projeto será necessário o uso de modelos de 30 amperes ou superiores.



Figura 3 - Modelo de Sensor de Corrente TC SCT 013-000 **Fonte:**
<http://s3.amazonaws.com/img.iluria.com/product/18B9DA/3A3A8A/450xN.jpg> (acesso em 24 de fevereiro de 2016).

Os sensores da família TC SCT são denominados não-invasivos, ou seja, como pode ser visto na Figura 3, ele detém de um encaixe lateral que se abre e fecha para a colocação do fio que precisa ser medido a corrente, este fato faz com que não se necessite qualquer manuseio da fiação oferecendo maior segurança na sua instalação.

4.6 *HyperText Markup Language* (HTML)

HyperText Markup Language, é o significado da sigla HTML, que, em português, significa linguagem para marcação de hipertexto. Ela foi criada por Tim Berners-Lee nos anos 90 tornando-se um padrão internacional. De modo geral, o hipertexto é todo o conteúdo de um documento para *web*, com a característica de se interligar a outros documentos da *web* através de links presentes nele próprio (SILVA, 2011).

Por este modo, o HTML não possui função de estilizar ou aplicar funcionalidades às páginas da *web*, apenas estruturando-as e fornecendo a elas princípios básicos de desenvolvimento que se adequam aos padrões definidos pela *web* (SILVA, 2011).

4.7 *Cascading Style Sheet* (CSS)

Cascading Style Sheet é o significado da sigla CSS, que, traduzido para o português, significa folhas de estilo em cascata. Tem a finalidade de definir estilos de apresentação para um documento HTML (SILVA, 2012).

Enquanto o HTML é destinado a fornecer uma estrutura para as páginas da *web*, este recurso tem o propósito de oferecer ao desenvolvedor uma estilização de fontes, textos,

imagens, espaçamentos, backgrounds, ou seja, qualquer marcação que exista dentro de um documento. Portanto, torna-se indispensável em qualquer aplicação web o uso desta tecnologia.

4.8 JavaScript

JavaScript é uma linguagem de programação criada pela Netscape em parceria com a Sun Microsystems. Sua primeira versão, definida como *JavaScript* 1.0, foi lançada em 1995 e implementada em março de 1996 no navegador Netscape Navigator 2.0 (SILVA, 2010).

Conforme Mozilla Developer Network (2015), *JavaScript* é baseado na linguagem de programação ECMAScript, o qual é padronizado pela Ecma International na especificação ECMA-262 e ECMA-402.

JavaScript será utilizado no projeto na fase de desenvolvimento da aplicação *web* que conterá a interface que o cliente/usuário irá acessar para verificação de todos seus dados lidos pelos sensores. A utilização de um *framework JavaScript* também será utilizado e detalhado logo abaixo, nas próximas seções.

4.9 AngularJS

AngularJS é um framework *open-source* mantido pela Google. Seu objetivo é estruturar o desenvolvimento *front-end* utilizando o modelo de arquitetura *model-view-controller* (MVC). O *framework* associa os elementos do documento HTML com objetos JavaScript, facilitando a manipulação dos mesmos.

Atualmente, existem duas versões do *framework*, a primeira e mais amplamente conhecida é a que será utilizada neste projeto, enquanto a outra lançada neste ano de 2016 contempla a utilização de outra forma de desenvolvimento que utiliza o *TypeScript*, que nada mais é que uma abstração da linguagem *JavaScript* para facilitar o desenvolvimento e se assemelhar à outras linguagens de programação como Java por exemplo, através da criação de componentes, existência de herança e variáveis com tipos definidos.

4.10 MYSQL

MYSQL é o banco de dados de código aberto mais popular do mundo e possibilita a entrega econômica de aplicativos de banco de dados confiáveis, de alto desempenho e redimensionáveis, com base na *web* e incorporados. (ORACLE, 2016).

Além da facilidade de uso, do alto desempenho e da confiabilidade do MySQL, pode-se beneficiar dos recursos avançados, das ferramentas de gerenciamento e do suporte técnico para desenvolver, implementar e gerenciar seus aplicativos.

Será amplamente utilizado no projeto para armazenar os dados lidos e todas as demais configurações e informações que precisem ser persistidas de forma *online*, como contas de usuário, históricos de acessos e outros.

4.11 NodeJS

Segundo Pereira (2013), NodeJS foi lançado em 2009 por Ryan Dahl com mais 14 colaboradores para solucionar um problema de bloqueio de execução em sistemas com muitas requisições simultâneas, isso fazia com que um processo em execução travava os outros até que se terminasse a execução do mesmo e assim os demais viessem a ser executados em fila.

“Esta tecnologia possui um modelo inovador, sua arquitetura é totalmente *non-blocking thread* (não-bloqueante), apresentando uma boa *performance* com consumo de memória e utilizando ao máximo e de forma eficiente o poder de processamento dos servidores[...]” (PEREIRA, 2013, p.2).

Esta tecnologia será utilizada no desenvolvimento do serviço *RestFul*, sendo implementada no *back-end* do projeto.

4.12 JavaScript Object Notation (JSON)

JSON é um formato de dados que pode ser trocado, permutado e que se assemelha nos padrões de *scripts*, porém, não é considerado um subtipo da sintaxe do *JavaScript* (MOZILA DEVELOPER NETWORK, 2014).

“ JSON é capaz de representar números, booleanos, textos, vazios, listas (sequência ordenada de valores) e objetos (mapeamento de valores de texto) composto por estes valores (ou por outras listas e objetos) ” (MOZILA DEVELOPER NETWORK, 2014, p1).

Um JSON está constituído em duas estruturas:

- Uma coleção de pares nome/valor.
- Uma lista ordenada de valores.

A aplicação deste formato no projeto será realizada para desacoplarmos nossas entidades de *hardware*, serviço e aplicação *web*, pois este formato é entendido facilmente por seres humanos e traduzidos com a mesma facilidade nos ambientes de máquinas, ou seja, podemos trocar totalmente qualquer entidade citada anteriormente por outra, até mesmo com outros equipamentos ou outras linguagens, bastando mantermos o formato JSON de entrada e saída de dados.

4.13 RESTFUL

Em 2000, Roy Fielding, em sua tese de doutorado apresentou uma nova forma de integrar sistemas hipermídias distribuído chamado REST (*Representational State Transfer*). (SANTOS, 2009)

Ainda de acordo com Santos (2009), REST é um estilo de arquitetura de software para sistemas hipermídia distribuídos, onde utilizamos um navegador web para acessar recursos, mediante a digitação de uma URL.

Esta arquitetura e conceito serão empregados no projeto na fase de desenvolvimento de uma API⁴ para padronizar e desacoplar nossas camadas de desenvolvimento que são: hardware, serviço e aplicação web. Esta arquitetura nos trará manutenibilidade do sistema e trará benefícios para futuras mudanças na arquitetura de hardware.

⁴ API - *Application Programming Interface* (Interface de Programação de Aplicativos). É a interface, normalmente documentada que uma biblioteca ou framework disponibiliza para o programador possa utilizá-la.

4.14 JQuery

JQuery é uma biblioteca baseada em *JavaScript* de *software* livre e aberto criado por John Resig e seu uso está sob as licenças do MIT e GPL, ou seja, está permitido que qualquer pessoa ou empresa use de seus recursos para fins pessoais como também profissionais (SILVA, 2014).

Em 2005 o autor começou a elaborar suas primeiras intenções sobre como resolver problemas de robustez de código para solução de simples problemas (SILVA, 2014). Foi então que em 2006 foi lançado o primeiro *plugin* para a biblioteca e disponibilizada de forma pública.

Ainda segundo Silva (2014), a função desta biblioteca é adicionar interatividade e dinamismo às páginas *web* que conseqüentemente irão proporcionar ao desenvolvedor, mecanismos necessários à criação de *scripts* fácil e usável, sem obstrução e causando no usuário final uma melhor experiência de usabilidade na web.

Esta biblioteca segue os padrões de desenvolvimento Web e é totalmente compatível com os novos recursos do atual CSS3. Porém, deve-se atentar em seu uso para que esteja nas conformidades das normas do W3C.

Seu uso no projeto tem como premissa a disponibilização de maior rapidez e usabilidade no desenvolvimento do código *front-end* que será destinado à aplicação para uso do cliente final da aplicação *web* proposta.

4.15 Materialize

Um *framework front-end* com *layout* moderno e responsivo que segue os padrões de desenvolvimento baseado em *Material Designer* da Google (MATERIALIZE, 2016).

Criado e projetado pela Google, o *Material Designer* é a combinação de princípios clássicos de projetos bem-sucedidos que contempla inovação e tecnologia, seguindo o objetivo de unificar a experiência do usuário em todos os seus produtos independente de qual plataforma está sendo utilizada. (MATERIALIZE, 2016).

Neste projeto irá ser implementado em toda aplicação que terá acesso dos usuários. Seu uso no desenvolvimento irá favorecer a usabilidade do cliente que contará com uma interface totalmente responsiva e inovadora.

5 REFERÊNCIAS

- ARDUINO Introduction. **Arduino**, 2016. Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em: 10 Março 2016.
- ASHTON, K. That "Internet of Things" Thing. **RFID Journal**, 2009. Disponível em: <<http://www.rfidjournal.com/articles/view?4986>>. Acesso em: 20 Março 2016.
- DAMAS, L. **Linguagem C**. 10. ed. Rio de Janeiro: LTC, 2007.
- EVANS, D. **The internet of things**: how the next evolution of the internet is changing everything. [S.l.]: CISCO white paper, 2011. 11 p.
- EVANS, M.; NOBLE, J.; HOCHENBAUM, J. **Arduino em ação**. São Paulo: Novatec Editora, 2013.
- INTRODUÇÃO ao Angular JS. **DEVMEDIA**, 2016. Disponível em: <<http://www.devmedia.com.br/introducao-ao-angularjs-curso-completo-de-angularjs-aula-1/32148>>. Acesso em: 10 Março 2016.
- KENDE, M. Global Internet report. **Geneva**: Internet Society, 2014. Disponível em: <<http://www.internetsociety.org/doc/global-internet-report>>. Acesso em: 21 Março 2016.
- KLOTZ, D. C for Embedded Systems Programming. **NXP**, 11 Novembro 2010. Disponível em: <http://www.nxp.com/files/training/doc/dwf/AMF_ENT_T0001.pdf>. Acesso em: 10 Março 2016.
- LACERDA, F.; LIMA-MARQUES, M. Da necessidade de princípios de Arquitetura da Informação para a Internet das Coisas. **Perspectivas em Ciência da Informação**, Belo Horizonte, 20, Junho 2015. 158-171.
- MATERIALIZE. Documentação - Materialize. **Materialcss**, 2016. Disponível em: <<http://materializecss.com/>>. Acesso em: 21 Março 2016.
- MCROBERTS, M. **Arduino básico**. São Paulo: Novatec Editora, 2011.
- MOZILA DEVELOPER NETWORK. **JSON**, 2014. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/JSON>>. Acesso em: 20 Março 2016.
- MYSQL | O Banco de Dados de Código Aberto Mais Popular. **ORACLE**, 2016. Disponível em: <<http://www.oracle.com/br/products/mysql/overview/index.html>>. Acesso em: 20 Fevereiro 2016.

PEREIRA, C. R. **Node.js**: Aplicações web real-time com Node.js. São Paulo: Casa do Código, 2015.

RASPBERRY PI, 2014. Disponível em: <<https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>>. Acesso em: 20 Março 2016.

SANTOS, W. R. D. **Univale**, 2016. Disponível em: <<http://www.univale.com.br/unisite/mundo-j/artigos/35RESTful.pdf>>. Acesso em: 23 Fevereiro 2016.

SCHILDT, H. **Linguagem C**: Guia do Usuário. São Paulo: MC Graw Hill, 1986.

SILVA, M. S. **JavaScript**: Guia do Programador. São Paulo: Novatec Editora, 2010.

SILVA, M. S. **HTML5**: A linguagem de marcação que revolucionou a web. São Paulo: Novatec Editora, 2011.

SILVA, M. S. **CSS3**: Desenvolva aplicações web profissionais com uso dos poderosos recursos de estilização das CSS3. São Paulo: Novatec Editora, 2012.

SILVA, M. S. **JQuery**: A Biblioteca do Programador JavaScript. 3. ed. São Paulo: Novatec Editora, 2014.

SOUZA, F. Primeiros Passos. **Arduino**, 2013. Disponível em: <<http://www.embarcados.com.br/arduino-primeiros-passos/>>. Acesso em: 22 Fevereiro 2016.

STAMFORD, C. Gartner Says the Internet of Things installed base will grow to 26 billion units by 2020. **Gartner**, 2013. Disponível em: <<http://www.gartner.com/newsroom/id/2636073>>. Acesso em: 21 Março 2016.

TAURION, C. **Software embarcado**: oportunidades e potencial de mercado. Rio de Janeiro: Brasport, 2005.

UPTON, E.; , G. H. **Raspberry pi**: Manual do usuário. Tradução de Celso Roberto Paschoa. São Paulo: Novatec Editora, 2013.