

UNIVERSIDADE DO VALE DO SAPUCAÍ
GUILHERME SANCHES PEREIRA
JÉSSICA ADRIELE DO NASCIMENTO

GERENCIADOR DE CONSUMO DE ENERGIA ELÉTRICA RESIDENCIAL

POUSO ALEGRE, MG

2016

UNIVERSIDADE DO VALE DO SAPUCAÍ

GUILHERME SANCHES PEREIRA

JÉSSICA ADRIELE DO NASCIMENTO

GERENCIADOR DE CONSUMO DE ENERGIA ELÉTRICA RESIDENCIAL

Trabalho de Conclusão de Curso apresentado ao Curso de Sistemas de Informação da Universidade do Vale do Sapucaí como requisito parcial para obtenção do título de bacharel em Sistemas de Informação.

Orientador: Prof. Artur Luis Ribas Barbosa

POUSO ALEGRE, MG

2016

LISTA DE FIGURAS

Figura 1 - Conexão entre as placas	6
Figura 2 - Conexão entre sensores e disjuntores	8
Figura 3 - Dados persistidos no banco online	9
Figura 4 - Último consumo de 16 de setembro às 12h e 09 minutos	9
Figura 5 - Último consumo de 16 de setembro às 12h e 10 minutos	9
Figura 6 - Painel de <i>login</i>	10
Figura 7 - Gráficos de apresentação de resultados	11
Figura 8 - Testes e validações com alicate amperímetro	11

SUMÁRIO

1	DISCUSSÃO DE RESULTADOS	6
	REFERÊNCIAS	12

1 DISCUSSÃO DE RESULTADOS

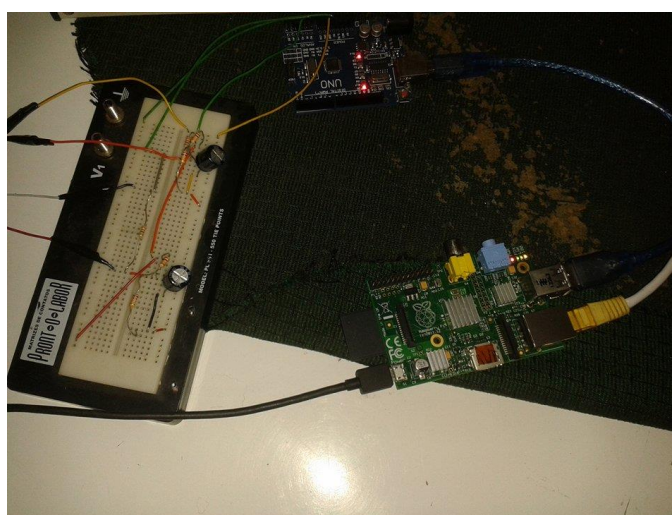
Neste capítulo são apresentados e discutidos os resultados obtidos pela pesquisa por meio do desenvolvimento das tecnologias empregadas no trabalho, com o intuito de demonstrar os pontos fortes e fracos encontrados.

Este trabalho teve como objetivo desenvolver uma aplicação *web* capaz de gerenciar o consumo de energia elétrica residencial. Esta aplicação tinha por objetivo medir e coletar a energia elétrica dos setores¹ de uma residência através de placas eletrônicas com sensores; armazenar os dados coletados em um banco de dados externo e demonstrar as informações coletadas para os usuários em um ambiente *online*.

Para alcançar os objetivos mencionados foram indispensáveis diversas pesquisas para modelagem da arquitetura utilizada e escolha das ferramentas e tecnologias que atenderiam da melhor forma as necessidades do trabalho.

A arquitetura desenvolvida contou com total desacoplamento entre as camadas refletindo resultados positivos em relação à manutenibilidade do sistema. Tendo sido definida a arquitetura e as tecnologias utilizadas foram feitas as ligações dos equipamentos seguidos de protótipos de produto final, como pode ser visto na figura X.

Figura 1 - Conexão entre as placas



Fonte – Elaborada pelos autores

¹ Definimos setores residenciais como sendo cada conjunto da residência que está conectado a um mesmo disjuntor elétrico. Alguns casos os setores representam: lâmpadas, tomadas e chuveiros; outros em: sala / cozinha, quartos e banheiros.

Com as conexões feitas, como pode ser visto na Figura 1 e funcionando corretamente o próximo passo foi desenvolver o código que seria gravado na plataforma de desenvolvimento Arduino. Este código teria por objetivo escrever na saída serial da Arduino o valor da corrente que circulava por uma extensão que utilizamos para teste. Ao término, os resultados tiveram variações se comparado ao valor especificado pelo fabricante de cada equipamento que havia sido ligado na extensão.

Com estes resultados insatisfatórios, pesquisou-se referências em outros trabalhos nesta área, o qual nos trouxe alternativas para tais problemas. Como havíamos nos apoiado em referências de outros trabalhos semelhantes que realizavam as mesmas medições, não notamos que os sensores utilizados em nossa pesquisa possuíam um circuito elétrico diferente dos quais haviam sido implementados pelos outros trabalhos, logo, foi necessário alterar o circuito elétrico para que o mesmo fosse capaz de atender o equipamento utilizado. Então os dados lidos começaram a apresentar resultados mais satisfatórios se comparado aos dos fabricantes, porém, ainda constavam variações maiores que 10% em alguns casos.

Este problema foi solucionado efetuando a devida calibração que o sensor de 100A necessitava como pode ser visto no Capítulo X sessão Y. Após os procedimentos apresentados, os resultados possuíam variação menores que 3%, que representa a variação máxima permitida pelo fabricante do sensor de corrente.

Tabela 1 – Medições comparativas dos equipamentos de teste

Dispositivo	Corrente A	Corrente B	Corrente C	Variação
Secador de cabelo	10	8,90	8,80	1,124%
Ferro de passar	10	9,35	9,23	1,283%
Ventilador	0,5	0,47	0,45	4,255%
Panela elétrica	3,15	3,14	3,1	1,274%
Chuveiro	15A	15,1	14,72	2,517%

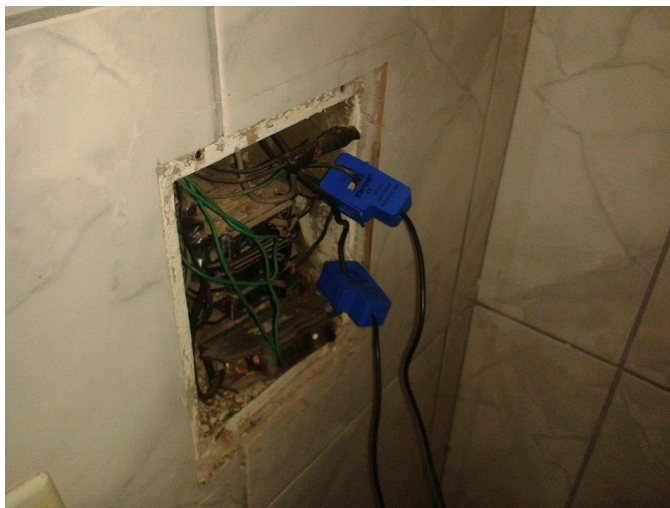
Corrente A	Fabricante	Media Variação
Corrente B	Alicate Amperímetro	2,0906%
Corrente C	Sensor SCT 013-000	Variação Fabricante
		3%

Fonte: Desenvolvida pelos autores

Como pode ser visto na Tabela 1, a média dos resultados obtidos apresentam uma porcentagem menor que a porcentagem que o fabricante do sensor disponibiliza.

A Figura X apresenta a forma que os sensores são ligados nos disjuntores.

Figura 2 - Conexão entre sensores e disjuntores



Fonte – Elaborada pelos autores

A ligação dos sensores ocorre de forma não invasiva, ou seja, não é necessária qualquer inclusão do sensor com a rede elétrica, o resultado apresenta uma forma mais prática e segura de se manusear com tensões elétricas.

Outro ponto que pode ser analisado com os resultados da Tabela 1 é que a menor corrente apresenta uma taxa de variação maior, este resultado se apoia na teoria de que como existe uma corrente de ruído que fica em torno de 0.1 a 0.4 amperes, as correntes menores que 1 ampere sofre maiores variações pois são somadas a estes ruídos. Logo, correntes maiores ao serem somadas com estes ruídos não representam dados significativos de variações.

Como os sensores são conectados a disjuntores elétricos, os mesmos tendem a sempre medirem correntes maiores que 1 ampere, por este fato, resultados satisfatórios nas leituras foram obtidos.

Após as leituras estarem dentro das taxas de variação aceitáveis, a persistência destes dados online foi obtida através da conexão da Raspberry com nossa API RESTful desenvolvida que recebe os dados na forma de POST e persistem os dados no banco de dados MySQL a cada 30 minutos.

Figura 3 - Dados persistidos no banco online

sensors_id	Users_id	devices_id	consumption	date
0	1	123456	0.357481	2016-09-14 17:51:49
1	1	123456	1.08963	2016-09-14 17:51:49
0	1	123456	0.357771	2016-09-14 18:21:45
1	1	123456	1.08992	2016-09-14 18:21:45
0	1	123456	0.357981	2016-09-14 18:51:50
1	1	123456	1.09036	2016-09-14 18:51:50
0	1	123456	0.358359	2016-09-14 19:21:50
1	1	123456	1.09068	2016-09-14 19:21:50

Fonte – Elaborada pelos autores

A Figura 1 demonstra os dados lidos entre as 17h51 às 19h:21 do dia 14 de setembro de 2016. Pode-se observar que existem 2 sensores conectados para o usuário de identificador 1 e placa Raspberry com identificador 12356 e o valor de consumo já é salvo na unidade de Killowatts/hora.

Outra conexão entre a Raspberry e a API é realizada a cada 20 segundos, atualizando o valor total até este momento em Killowatts/hora. Como pode ser visto na Figura 2 e Figura 3.

Figura 4 - Último consumo de 16 de setembro as 12h e 09 minutos

user_id	last_consumption	date
1	7.104446030500022	2016-09-16 12:09:52

Fonte – Elaborada pelos autores

Figura 5 - Último consumo de 16 de setembro as 12h e 10 minutos

user_id	last_consumption	date
1	7.104441032000023	2016-09-16 12:10:13

Fonte – Elaborada pelos autores

A Figura 2 apresenta o valor total consumido em killowatts/hora pelos 2 sensores até as 12 horas 09 minutos e 52 segundos do dia 16 de setembro e a Figura 3 após 20 segundos da última medição.

O tempo da requisição pode variar de acordo com a internet local e este tempo representar uma variação na data salva no banco de dados, a Figura 2 e a Figura 3 apresentam uma variação de tempo de 21 segundos, sendo que são programados para 20 segundos. Porém este resultado não interfere no objetivo final da pesquisa.

O último resultado se refere à apresentação dos dados lidos e persistidos de forma online ao usuário final através de uma interface *web* a qual o usuário possui um usuário e senha que o permite realizar login de forma segura. Como pode ser visto na Figura 4.

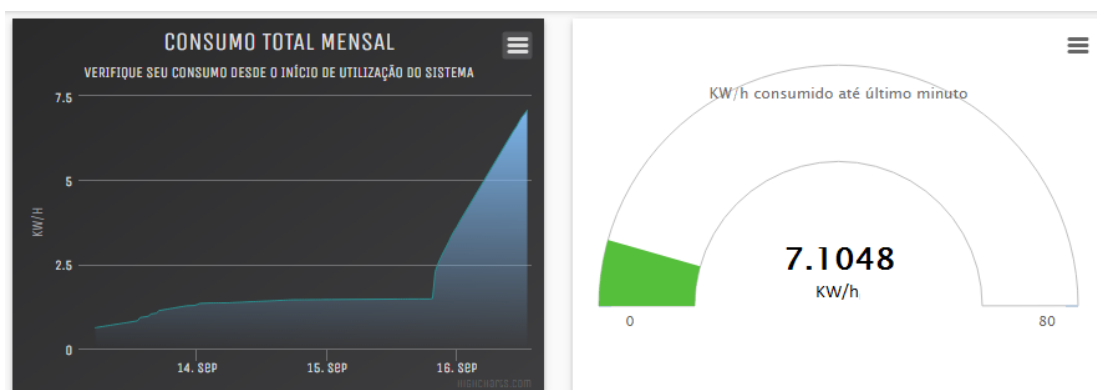
Figura 6 - Painel de *login*

Fonte – Elaborada pelos autores

A forma de login segue padrões de segurança através de *token authentication* que expira a cada dia. Esta validação da API assegura que usuários mal-intencionados tenham maiores dificuldades caso queiram acessar dados dos usuários desta solução proposta.

Após realizarem *login* já são apresentados aos resultados através de gráficos que representam valores em kilowatts/hora desde o início da implantação do sistema, um gráfico que se assemelha a um velocímetro que marca o total de kilowatts/hora consumidos até os últimos 20 segundos.

Figura 7 - Gráficos de apresentação de resultados

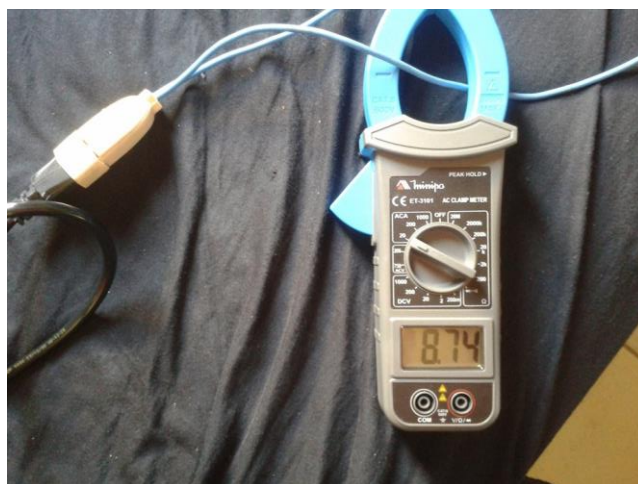


Fonte – Elaborada pelos autores

A Figura 5 mostra os 2 gráficos representando os valores lidos até os últimos 20 segundos e os dados desde o início do monitoramento.

Pode-se notar na Figura 5, no primeiro gráfico que houve um pico ao final do dia 16 de setembro de 2016, este pico ocorreu, pois, antes deste período, eram realizados os testes através de uma extensão que os equipamentos eram ligados e os valores de corrente validados através da mensuração com alicate amperímetro, como pode ser visto na Figura Z.

Figura 8 - Testes e validações com alicate amperímetro



Fonte – Elaborada pelos autores

Após este período de testes e validações, os sensores foram conectados aos disjuntores da residência e então os valores ficaram mais altos devidos a estarem mensurando agora toda a parte elétrica da residência.

REFERÊNCIAS

AGUILAR, L. J. **Fundamentos de Programação - Algoritmos, estruturas de dados e objetos**. 3ª Edição. ed. São Paulo: AMGH Editora, 2008.

ANGULARJS. O que é AngularJS. **ANGULARJS**, 2016. Disponível em: <<https://docs.angularjs.org/guide/introduction>>. Acesso em: 11 Agosto 2016.

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.

CAELUM. Rest - Web Ágil com VRaptor, Hibernate e Ajax. **CAELUM**, 2016. Disponível em: <<https://www.caelum.com.br/apostila-vraptor-hibernate/rest/#11-3-o-triangulo-do-rest>>. Acesso em: 02 Agosto 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.

FREMAN, A. **Pro AngularJS**. Nova Iorque: Apress Media LLC, 2014.

GREEN, B.; SESHADRI, S. **AngularJS**. Sebastopol: O'Reilly Media, 2013.

HIGHCHARTS. Highcharts Documentation. **Highcharts**, 2016. Disponível em: <<http://www.highcharts.com/docs>>. Acesso em: 26 Julho 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.

MONGODB. MongoDB for GIANTI Ideas. **MongoDB**, 2016. Disponível em: <<https://www.mongodb.com/>>. Acesso em: 01 Agosto 2016.

MOREIRA, R. H. O que é Nodejs? **NodeBr**, 2013. Disponível em: <<http://nodebr.com/o-que-e-node-js/>>. Acesso em: 20 Julho 2016.

MOZILLA DEVELOPER NETWORK. **JSON**, 2014. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/JSON>>. Acesso em: 20 Março 2016.

MOZILLA DEVELOPER NETWORK. **JavaScript**, 2016. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>>. Acesso em: 22 Abril 2016.

ORACLE. 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**, 2009. 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. **JQuery**: A Biblioteca do Programador JavaScript. 3. ed. São Paulo: Novatec Editora, 2014.

SILVEIRA, G. Internet das Coisas - O que é IoT? **Bluelux**, 2016. Disponível em: <<http://www.bluelux.com.br/internet-das-coisas-iot/>>. Acesso em: 31 Julho 2016.

SOARES, J. O que é MongoDB e porque usá-lo? **Código Simples**, 2016. Disponível em: <<http://codigosimples.net/2016/03/01/o-que-e-mongodb-e-porque-usa-lo/>>. Acesso em: 10 Agosto 2016.

SOUZA, F. Arduino UNO - Conheça os detalhes de seu hardware. **Embarcados**, 2016. Disponível em: <<http://www.embarcados.com.br/arduino-uno/>>. Acesso em: 20 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.

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