



## SISTEMA DE CONTROLE DIGITAL PARA MONITORAMENTO E GERENCIAMENTO DO CONSUMO ENERGÉTICO DE UMA RESIDÊNCIA

Cleiber Nichida\*<sup>1</sup>, Dannylo M. Guerra<sup>1</sup>, Rodrigo F. Gonçalves<sup>1</sup>, João B. J. Pereira<sup>1</sup>

<sup>1</sup>IFG – Instituto Federal de Educação, Ciência e Tecnologia de Goiás

**Resumo** – Controlar e monitorar circuitos elétricos tornou-se uma necessidade primária para a tomada de decisões que visam melhorias na eficiência energética. Este trabalho propõe o desenvolvimento de um sistema de controle automatizado para residências, também conhecido como DCR (*Digital Control for Residence* – Controle Digital para Residência). Através de seu uso, a utilização da energia elétrica em uma residência pode ser observada e controlada por um dispositivo conectado à rede LAN/WLAN (*Local Area Network/Wireless Local Area Network* - Rede Local Sem Fio) da casa ou à internet de qualquer lugar. A princípio, foi desenvolvido um sistema capaz de ser executado por qualquer navegador ou por dispositivos móveis como *tablets* e celulares que utilizem o sistema operacional Android ou IOS, através do aplicativo *Home Assistant* ou do próprio navegador do dispositivo. Algumas vantagens apresentadas neste projeto são a mobilidade, comodidade, segurança, bem-estar e a utilização de um recurso extremamente eficaz e com baixo consumo de energia para gerenciamento de toda a aplicação.

**Palavras-Chave** - Automação Residencial, Controle, Eficiência Energética, Internet, Raspberry PI.

### DIGITAL CONTROL SYSTEM FOR MONITORING AND MANAGING THE ENERGY CONSUMPTION OF A RESIDENCE

**Abstract** - Controlling and monitoring electrical circuits has become a primary need for decision-making aimed at improving energy efficiency. This work proposes the development of an automated home control system, also known as DCR (*Digital Control for Residence*). Through its use, the electricity usage in a home can be observed and controlled by a device connected to the home's LAN/WLAN (*Local Area Network/Wireless Local Area Network*) or to the internet from anywhere. At first, a system was developed capable of being executed by any browser or by mobile devices such as tablets and cell phones that use the Android or IOS operating system, through the *Home Assistant* application or the device's own browser. Some advantages presented in this project are mobility, convenience, safety, well-being and the use of

an extremely efficient and low-energy resource for managing the entire application.

**Keywords** - Control, Energy Efficiency, Home Automation, Internet, Raspberry PI.

### I. INTRODUÇÃO

A automação residencial vem há décadas deixando de ser apenas um conceito e tornando-se realidade. É um dos principais caminhos trilhados atualmente dentro do cenário da modernização tecnológica, a qual vários processos estão sendo automatizados com a finalidade de garantir maior comodidade e melhor desempenho, em que se tratando de consumo energético, a maximização da performance está pautada na eficiência energética.

A substituição de equipamentos de grande consumo de eletricidade por outros mais eficientes viabiliza, em termos tecnológicos, uma utilização eficiente de energia [1]. Entretanto, esta medida não impede que os equipamentos sejam utilizados de uma maneira inadequada.

A Domótica é a utilização simultânea da eletricidade, da eletrônica e das tecnologias da informação no ambiente residencial, permitindo realizar a sua gestão, local ou remota, e oferecer uma vasta gama de aplicações integradas nas áreas da segurança, conforto, comunicação e gestão de energia [2]. Essa inovação está cada vez mais se traduzindo em um diferencial na opção de compra de novas habitações, agregando valor às mesmas [3]. Nesses ambientes, a Computação Ubíqua ou “invisível” é uma ferramenta bastante utilizada na assistência ao usuário e mesmo na análise do funcionamento das mesmas [4]. A Computação Ubíqua é um termo que foi cunhado originalmente por Mark Weiser em 1991 para se referir a dispositivos conectados de forma tão transparente para o ser humano que se torna imperceptível aos mesmos [5].

O emprego de Redes Elétricas Inteligentes (*Smart Grid*) é uma atividade que se insere no contexto, atrelando eficiência energética à automatização, sendo um modelo buscado principalmente por consumidores residenciais de baixa tensão, os quais objetivam obter uma execução de processos com uma menor demanda de energia. Sob essa ótica, o estabelecimento de medidas que visam uma maior eficiência energética no Brasil se tornam necessárias, e é a partir desse quadro que surge a aplicação das Redes Elétricas Inteligentes [6].

\*cnichida@gmail.com

Para que todo o conceito de Redes Elétricas Inteligentes seja empregado, um dos principais elementos para o seu funcionamento é o *Smart Meter*, em que a partir dele é possível a obtenção de dados de diferentes espécies, relacionados às características da energia elétrica distribuída e consumida [7]. Esse gerenciamento pode ser feito por um SGE (Sistema de Gerenciamento de Energia) que integra microprocessadores, painéis, cabos de comunicação e programas de aplicação de software.

A automação residencial traz muitos benefícios aos usuários. Contudo, seu custo de implantação é alto, e um dos motivos para esse alto custo é a necessidade de hardware dedicado para o controle residencial. Para reduzir o valor da implementação da automação residencial, é possível usar uma categoria de hardware mais recente chamada OSH (*Open-Source Hardware - Hardware de Código Aberto*) ou hardware livre no mundo da computação.

Nesse contexto, surgem ambientes inteligentes caracterizados por permitir a interação do usuário com os dispositivos espalhados por todo o local no qual ele se encontra, propiciando além da interação usuário-dispositivo, a interação entre os próprios dispositivos e abrindo espaço para a personalização das suas ações de acordo com comportamentos previsíveis [8]. Assim, abre-se a possibilidade de o usuário customizar esses dispositivos não unicamente com suas preferências, mas aplicando técnicas voltadas a melhoria de sua qualidade de vida e produtividade.

O presente estudo tem como foco o desenvolvimento de um sistema controlador digital para residência, de baixo custo, que traz a esse contexto, melhorias que buscam eficiência ao sistema e controle para o usuário. Tal controlador funciona de modo remoto, podendo ser utilizado tanto pela rede local quanto pela *internet*, sendo que a interface pode ser utilizada tanto através de um computador quanto por um *smartphone*.

Especificamente neste caso, o usuário doméstico passa a ter informações claras sobre seus hábitos de consumo energético para que ele utilize mais racionalmente a eletricidade, de forma simples e acessível. Todavia, uma vez aperfeiçoado, o sistema desenvolvido poderá também contribuir para a modernização de uma residência com baixos investimentos, tendo como consequência diversos benefícios, tais como: redução de gastos desnecessários com energia elétrica, análise do desempenho e funcionamento dos eletrodomésticos, controle à distância de iluminação e equipamentos, dentre vários outros, de acordo com a prática que o usuário vá adquirindo com o seu uso.

## II. MÉTODOS E MATERIAIS

Para implantação do trabalho prático, após a pesquisa bibliográfica e das folhas de dados dos componentes, elaborou-se o projeto do dispositivo, e uma bancada foi construída para simular uma residência com alguns pontos de tomadas, buscando de tal forma que fosse um cenário possível para utilização do DCR, como mostram a Figura 1, seu descritivo na Tabela I e seu diagrama elétrico na Figura 2.

Os microcontroladores utilizados neste trabalho foram o *Raspberry Pi* de terceira geração, modelo B cuja função principal foi a de propiciar a interação entre a coleta de dados e banco de dados, gerando ainda a interface do sistema com o usuário, facilitando assim a interpretação dos dados. O

*Arduino* foi utilizado para gerenciar os dados dos atuadores e sensores e enviá-los para o *Raspberry Pi* através de um cabo de dados serial. O modelo escolhido foi o *Mega 2560* por sua robustez.

Figura 1: Bancada montada para simulação.

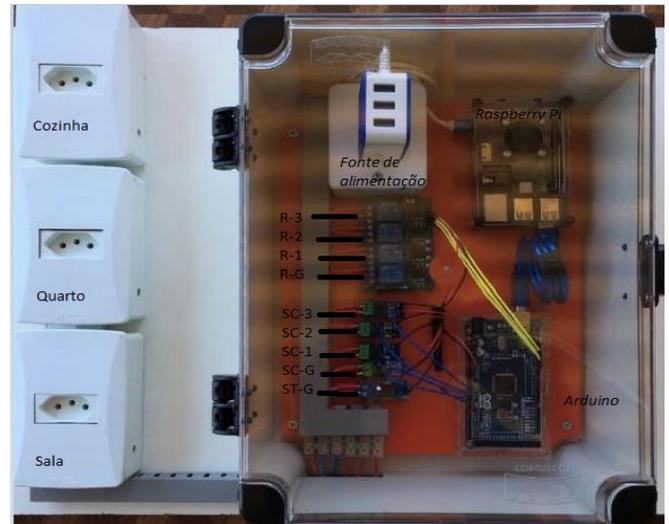
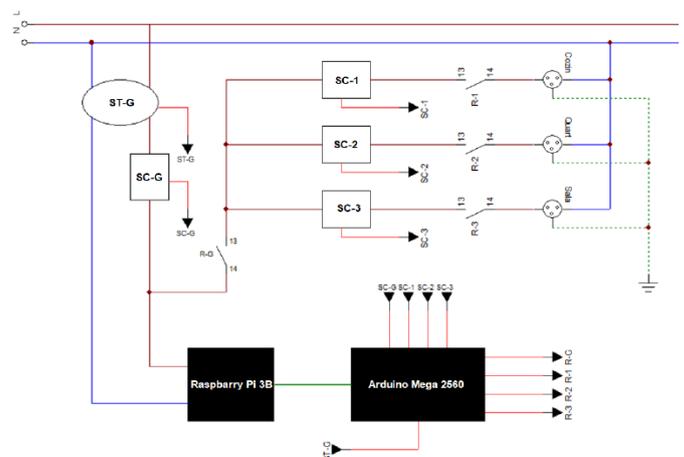


Tabela I: Legenda da Figura 1.

Sigla	Descrição
ST-G	Sensor de Tensão Geral
SC-G	Sensor de Corrente Geral
SC-1	Sensor de Corrente Circuito 01
SC-2	Sensor de Corrente Circuito 02
SC-3	Sensor de Corrente Circuito 03
R-G	Relé Geral
R-1	Relé Circuito 01
R-2	Relé Circuito 02
R-3	Relé Circuito 03
Cozinha	Tomada Cozinha
Quarto	Tomada Quarto
Sala	Tomada Sala

Figura 2: Diagrama elétrico da bancada.



Por se tratar de um projeto experimental e educacional, os relés estão sendo os *switches* diretos para ligar e desligar o circuito. No entanto, para uma aplicação em planta real, é aconselhável que os relés sejam tratados como responsáveis pelo acionamento das bobinas de contadores por estas exigirem

baixa corrente para seu acionamento, mas que possuem capacidade de suportar correntes maiores que 10 A em seus contatos principais.

Na Tabela II, listou-se os materiais e recursos utilizados na confecção deste protótipo.

Tabela II: Lista de materiais utilizados.

Item	Função
Raspberry Pi 3 model B	Propiciar a interação entre a coleta de dados e o banco de dados, gerando também a interface do sistema com o usuário através do <i>Node-RED</i> e o usuário, facilitando a interpretação dos dados e tornando o sistema mais amigável.
Arduino Mega 2560	Coleta dos dados dos sensores e controle dos relés.
Sensores de corrente (ACS712) / Sensores de tensão (AC P8)	Responsáveis pela coleta de dados de corrente e tensão nos circuitos
Relay Module	Acionado pelo <i>Arduino</i> , funciona como um interruptor eletromecânico tendo a vantagem de utilizar um acoplador óptico garantindo que não haja nenhum contato elétrico entre o circuito de acionamento do relé e o <i>Arduino</i> .
Node-RED	Ferramenta visual de ambiente de código aberto que por meio dos <i>nodes</i> ou nós, é possível ler arquivos CSV, escutar eventos http, tcp, <i>websocket</i> , <i>twitter</i> , <i>mqtt</i> entre outros.
Grafana	plataforma para visualizar e analisar métricas por meio de gráficos.
Home Assistant	Provê a interação do usuário com o sistema de automação.
Plataforma de madeira 60x40cm	Base da bancada de simulação.
Módulos de tomada tipo sobrepor	Simulam os pontos de energia de uma residência.
Caixa de montagem plástica transparente	Utilizada para a montagem do módulo de controle
Nabu Casa	Sistema pago que permite o acesso remoto ao sistema através de link externo pela alocação de um DNS.

### A. Preparação do controlador central.

Foi instalado o sistema operacional Hass.io (*Home Assistant*) no *Raspberry Pi* através de um cartão de memória pelo seu arquivo de instalação, disponibilizado no próprio site do *Home Assistant*. O endereço para o acesso da plataforma utilizado: <http://homeassistant.local:8123>. Após a instalação e toda configuração inicial realizada, necessitou-se da instalação de *Add-ons*, softwares complementares específicos para funções diferentes e que auxiliam o uso da automação nas residências. Neste projeto foram instalados os seguintes *Add-ons*:

- *File Editor*: complemento que permite adicionar um editor de arquivos baseados em navegador, permitindo a programação do *Home Assistant* em sua tela principal;
- *SambaShare*: complemento que compartilha as pastas e arquivos que estão dentro do sistema operacional com outros computadores da rede local, permitindo o envio de arquivos;
- *Node-RED*: serviço *open source* que permite realizar funções e programações baseados nos dados recebidos e então organizar os dados de acordo com o objetivo do projeto;

- *Influx DB*: serviço *open source* de banco de dados de séries temporal, que possui uma capacidade de armazenamento de 10GBs;
- *Grafana*: serviço *open source* com uma multiplataforma e visualização interativa da *web*. Ele fornece tabelas, gráficos e alertas.
- *NabuCasa*: serviço responsável por compartilhar a plataforma gerenciada do *Home Assistant* para um link externo, permitindo acesso de todo local com acesso à *internet*.

### B. Preparação do microcontrolador.

A linguagem de programação do *Arduino* pode ser dividida basicamente em três partes, sendo: estruturas, valores (variáveis e constantes) e funções. Para o código destacam-se entre as principais funções utilizadas: funções de entradas e saídas de dados digitais e analógicos, funções temporizadoras e funções de comunicação.

Basicamente todos os parâmetros utilizados no projeto foram do tipo *float* (flutuante) por se tratar de grandezas elétricas com valores reais. Logo, todos os valores alvo do projeto (tensão, corrente e potência), foram declarados sendo do tipo *float*.

Quanto às estruturas, utilizou-se neste código funções de controle tais como *for* e *if*, alguns operadores de comparação do tipo “<=” (menor ou igual) e “==” (igual) e ainda elementos de sintaxe (*#define*, *#include*...), dentre outros.

A principal biblioteca utilizada para o código foi a “ACS712.h”, com a qual de modo fácil e simples, controlam-se os sensores de corrente ACS712 30A utilizados para a coleta dos dados referentes às correntes elétricas. Esta biblioteca permite também efetuar a calibração dos sensores sempre que solicitado mantendo sempre uma leitura acurada.

O código foi dividido em 4 partes:

- *Declarações* – são declaradas sempre no começo, todas as variáveis e bibliotecas a serem utilizadas no programa. A linguagem não permite a utilização de qualquer variável sem sua devida declaração com o tipo de dado que será utilizado/recebido por aquela variável. As bibliotecas devem ser definidas também nesse estágio do programa para que sejam iniciadas imediatamente quando a placa do *Arduino* for alimentada e o *script* (código) inicializado.
- *Setup* – a estrutura *setup* é responsável por inicializar variáveis, configurar o modo de funcionamento de pinos (entrada ou saída de dados), inicializar bibliotecas etc. É executada uma única vez após a placa ser alimentada ou ocorrendo um *reset*.
- *Loop* – a estrutura *loop* faz precisamente o que seu nome sugere: repete consecutivamente o código em seu interior enquanto a placa estiver ligada. É onde ocorre a leitura de dados do projeto, recebe e envia informações pela porta serial etc.
- *Envio de dados* – momento ainda dentro da estrutura *loop* no qual após a coleta de todos os dados necessários em cada ciclo, envia os dados pela porta serial, para que seja alimentado o banco de dados.

O código programado propõe a coleta dos valores de tensão em um ponto da rede elétrica, os valores de corrente geral

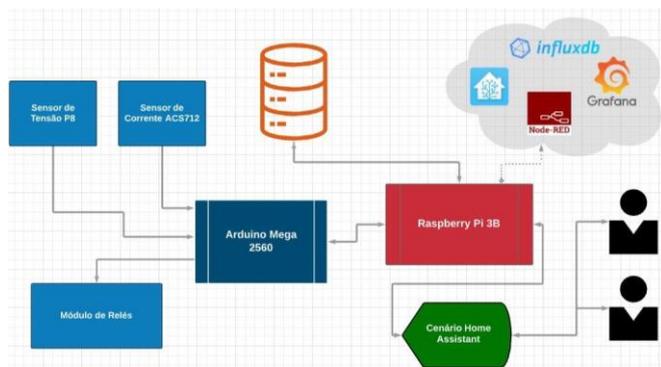
fornecida para o circuito e os valores da corrente consumida para cada circuito. Com estes dados calculam-se através de operadores aritméticos a potência instantânea consumida pelo circuito específico e a potência consumida pelo circuito geral.

### C. Fluxo de processos

Com a central de controle e o microcontrolador devidamente configurados e realizando comunicação entre si, é iniciado-se o processo de tratativa dos dados para que chegassem até o usuário. Seguiram-se os seguintes processos:

- 1) Sensores de corrente e tensão fornecendo dados de entrada para o microcontrolador;
- 2) Arduino em *loop* constante faz a leitura dos dados, os separa e envia para a central de controle;
- 3) O *Raspberry Pi* ao receber os dados começa a realizar o gerenciamento, onde alguns serão armazenados no banco de dados e outros já passarão a ser exibidos em tempo real no cenário do *Home Assistant*;
- 4) O banco de dados da série temporal armazena todos os valores e os disponibiliza para o *Grafana* apresentá-los em forma de gráficos para o usuário;
- 5) O cenário do *Home Assistant* está configurado com *dashboards* para exibição dos valores em tempo real, para filtragem e análise dos dados e para controle do usuário, permitindo-o a inserir dados de entrada como comandos para ligar e desligar o circuito;
- 6) O usuário consegue acesso ao sistema de qualquer lugar, desde que possua acesso à *internet*, e então consegue monitorar e controlar todo o projeto da forma que preferir;
- 7) Caso o usuário forneça algum comando, o sinal passará pela central e pelo microcontrolador até chegar ao módulo de relés, onde estes atuarão de acordo com o comando.

Figura 3: Fluxo de processo DCR.



Na Figura 3, é ilustrado todo o fluxo de processos do DCR, desde a coleta nos sensores até a chegada dos dados ao armazenamento e ao usuário, onde o mesmo consegue visualizar e interagir através do cenário do *Home Assistant*.

### D. Cenário do Home Assistant.

Na Figura 4 é possível visualizar o *layout* do cenário disponibilizado no *Home Assistant* no qual o usuário terá acesso para monitorar e controlar o circuito. Cabe ressaltar que o *layout* apresentado é de um usuário com permissões de administrador.

Figura 4: Cenário *Home Assistant*.



São observados no cenário:

- 1) Barra do menu principal: Nela é possível navegar por toda interface do *Home Assistant*, desde a página inicial até os pontos de configuração de todo o projeto;
- 2) Barra de menu auxiliar: Opções inseridas na página inicial para deixar em fácil acesso informações que são fundamentais para o usuário.
- 3) Card de comandos: Um card é um bloco no qual é programado algumas funções específicas. Neste há apenas switches disponíveis para o usuário estar controlando o circuito;
- 4) Card histórico de estado: Este mostra de modo bem ilustrativo e intuitivo quanto tempo cada circuito permaneceu em um estado, seja ligado ou desligado;
- 5) Card para tensão: Um dashboard em formato de ponteiro para apresentar os valores da tensão da rede na qual o projeto está vinculado;
- 6) Card para corrente: Um dashboard em formato de ponteiro para apresentar os valores de corrente de cada circuito no qual o projeto está vinculado;
- 7) Card para potência: Um dashboard em formato de ponteiro para apresentar os valores de potência de cada circuito no qual o projeto está vinculado;

### E. Programação Arduino

Para melhores resultados e maior precisão dos valores mensurados, realizaram-se laços de repetição na coleta de corrente e tensão e posteriormente calculada a média aritmética simples da quantidade de dados lidos a cada laço. Para corrente aplicou-se apenas 10 laços por já utilizar uma biblioteca robusta para controle dos sensores. Para a leitura de tensão 1000 laços, ou seja, realizou-se 1000 leituras a cada loop do código para que o resultado obtido seja o mais preciso o possível. Mesmo com tantas leituras, devido a velocidade de processamento do *Arduino* escolhido (16 MHz), cada loop foi programado para ser realizado a cada 2 segundos em média.

Ao final de cada loop os dados coletados e armazenados em suas devidas constantes, são enviados via porta serial, para o *Raspberry Pi* no qual armazenará os dados no banco de dados. Os dados enviados são (na sequência): Tensão, Corrente Geral, Corrente da Cozinha, Corrente da Sala, Corrente do Quarto, Potência Geral, Potência da Cozinha, Potência da Sala e Potência do Quarto. O código também tem a função de monitorar a porta serial a todo momento a espera de dados

recebidos oriundos do *Raspberry Pi* no qual irá comandar os relês de controle dos circuitos.

Na Figura 5, temos a programação do *Arduino*. Detalhe importante para as funções *Serial.Available*, *Serial.Read* e *Serial.Write*, pois essas são as responsáveis por fazer a comunicação com o *Raspberry Pi*.

Figura 5. Código-fonte do *Arduino*

```

Código Arduino

//*****Declaração de bibliotecas e variáveis*****
#include <ACS712.h>
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
ACS712 sensorGeral(ACS712_30A, A0);
ACS712 sensorCocinha(ACS712_30A, A1);
ACS712 sensorSala(ACS712_30A, A2);
ACS712 sensorQuarto(ACS712_30A, A3);
float Tensao = 0;
float I_Geral = 0;
float ICocinha = 0;
float ISala = 0;
float IQuarto = 0;
float PotenciaGeral = 0;
float PotenciaCocinha = 0;
float PotenciaSala = 0;
float PotenciaQuarto = 0;
int relG = 7;
int relCocinha = 8;
int relSala = 9;
int relQuarto = 10;
int amareles;

//*****Medição dos Correntes*****
int intercorrente = 5;
ICorrente = 0;
ISala = 0;
IQuarto = 0;
for(int contG = 0; contG < intercorrente; contG++){
  sensorGeral.calibrar();
  sensorCocinha.calibrar();
  sensorSala.calibrar();
  sensorQuarto.calibrar();
  delay(1);
  float MediaCorrenteGeral = sensorGeral.getCurrentAC(60);
  I_Geral = I_Geral + MediaCorrenteGeral;
  float MediaCorrenteCocinha = sensorCocinha.getCurrentAC(60);
  ICocinha = ICocinha + MediaCorrenteCocinha;
  float MediaCorrenteSala = sensorSala.getCurrentAC(60);
  ISala = ISala + MediaCorrenteSala;
  float MediaCorrenteQuarto = sensorQuarto.getCurrentAC(60);
  IQuarto = IQuarto + MediaCorrenteQuarto;
  delay(1);
}
ICorrente = I_Geral/intercorrente;
if(ICorrente >= 0.08){
  I_Geral = 0;
}
ICocinha = ICocinha/intercorrente;
if(ICocinha >= 0.08){
  ICocinha = 0;
}
ISala = ISala/intercorrente;
if(ISala >= 0.08){
  ISala = 0;
}
IQuarto = IQuarto/intercorrente;
if(IQuarto >= 0.08){
  IQuarto = 0;
}

//*****Medição de Tensão*****
float mediatensao = 0;
int intertensao = 1000;
for(int cont = 0; cont < intertensao; cont++){
  float voltagem = analogRead(A4);
  mediatensao = mediatensao + voltagem;
  delay(1);
}
mediatensao = mediatensao/intertensao;
Tensao = (mediatensao*5)/1023*46.7; //Fator de correção da tensão

//*****Cálculo das Potências*****
PotenciaGeral = Tensao * I_Geral;
PotenciaCocinha = Tensao * ICocinha;
PotenciaSala = Tensao * ISala;
PotenciaQuarto = Tensao * IQuarto;

//*****Tela LCD Interface*****
lcd.begin(16, 2);
if(PotenciaGeral >= 9999){
  lcd.print("Pot = "); lcd.print(PotenciaGeral); lcd.print("W");
}
else {
  float kW = PotenciaGeral/1000;
  lcd.print("Pot = "); lcd.print(kW); lcd.print("kW");
}

//*****Entrada de dados via Serial*****
Serial.print("Tensao:"); Serial.print(" ");
Serial.print("Geral:"); Serial.print(" ");
Serial.print("Cocinha:"); Serial.print(" ");
Serial.print("Sala:"); Serial.print(" ");
Serial.print("Quarto:"); Serial.print(" ");
Serial.print("PotenciaGeral:"); Serial.print(" ");
Serial.print("PotenciaCocinha:"); Serial.print(" ");
Serial.print("PotenciaSala:"); Serial.print(" ");
Serial.print("PotenciaQuarto:"); Serial.print(" ");
}

//*****Loop*****
void loop() {
  //*****Controle dos relês*****
  if (Serial.available()) {
    amareles = Serial.read();
    //*****Geral com redundância*****
    if (amareles == 49) {
      digitalWrite(relG, HIGH);
      digitalWrite(relCocinha, HIGH);
      digitalWrite(relSala, HIGH);
      digitalWrite(relQuarto, HIGH);
    }
    if (amareles == 48) {
      digitalWrite(relG, LOW);
      digitalWrite(relCocinha, LOW);
      digitalWrite(relSala, LOW);
      digitalWrite(relQuarto, LOW);
    }
    //*****Cocinha*****
    if (amareles == 51) {
      digitalWrite(relCocinha, HIGH);
    }
    if (amareles == 50) {
      digitalWrite(relCocinha, LOW);
    }
    //*****Sala*****
    if (amareles == 53) {
      digitalWrite(relSala, HIGH);
    }
    if (amareles == 52) {
      digitalWrite(relSala, LOW);
    }
    //*****Quarto*****
    if (amareles == 55) {
      digitalWrite(relQuarto, HIGH);
    }
    if (amareles == 54) {
      digitalWrite(relQuarto, LOW);
    }
  }
}

```

O link do código está disponível no endereço: <https://github.com/engnichida/codigoarduino>

### F. Programação Node-RED.

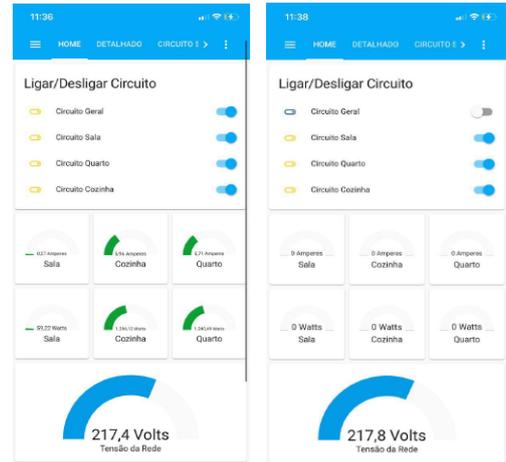
Os *nodes* principais utilizados no projeto foram: *Serial in*, *serial out*, *function*, *entity*, *influxdb out*, *debug*, *switch* e *split*. Um dos principais *nodes* é o *function*, onde contém uma programação *JavaScript* responsável por filtrar os dados recebidos pelo *Arduino* e então dar destino a eles para o cenário do usuário.

## III. RESULTADOS

Os resultados obtidos após os testes foram todos satisfatórios. O circuito foi analisado tanto em rede local quanto em rede externa, através do navegador e do aplicativo instalado em *smartphones*. Todos os comandos, tanto ativados localmente quanto remotamente (até mesmo em dispositivos de comando localizados em outra unidade da federação), funcionaram satisfatoriamente, apresentando apenas um atraso devido ao tempo de resposta da rede.

Na Figura 6 é possível visualizar na primeira imagem, a apresentação dos resultados ao usuário em uma tela de *smartphone*. Todos os circuitos estão ativados e os *dashboards* estão mostrando os valores de Corrente, Potência e Tensão respectivamente, nos circuitos onde estão ligados equipamentos às tomadas. Na imagem ao lado, o circuito geral foi desativado através do comando no aplicativo e todos os circuitos foram desligados.

Figura 6: Tela em *smartphone* rodando o aplicativo DCR.



Na Figura 7 é visualizado como os dados são exibidos em forma de gráfico para que o usuário possa fazer diversas análises no tempo em relação ao consumo.

Figura 7: Tela em *smartphone* – resultados em gráficos Tela inicial do navegador de um computador.



A visualização em um computador, utilizando um navegador é mostrada na Figura 8, onde temos a caixa de comandos e as caixas de visualização de dados. Através da visualização dos gráficos, possibilitou-se a extração de dados tais como a média de consumo em determinado período e o valor total. Um detalhe observado, como ponto negativo, foi que ao tentar filtrar dados de longos períodos, o processador do *Raspberry Pi* apresentou aquecimento excessivo, deixando-o com grande lentidão e em alguns casos chegando até a reiniciar como medida de segurança própria. Tal evento foi corrigido com a instalação de uma case para o aparelho. Essa case possui tanto ventoinhas quanto dissipadores,

facilitando de tal forma o resfriamento dos componentes durante o processamento de grandes volumes de dados.

Figura 8. Tela inicial do navegador de um computador.



Para conferência dos valores lidos pelo projeto foi utilizado um multímetro Fluke 177 True RMS, com certificado de calibração N° R1844/20 emitido em dezembro de 2020 de acordo com a NBR ISO/IEC 17025. A diferença dos valores entre o projeto e o multímetro chegou a ser de no máximo 5%, as variações se davam pelo tipo de carga utilizada para teste. Tal valor foi considerado satisfatório para o projeto executado.

#### IV. CONCLUSÃO

A automação residencial é uma área que certamente continuará a ser explorada. Este artigo apresentou a utilização de elementos OSH para automação residencial e teve o objetivo de apresentar dispositivos computacionais como *Raspberry Pi* e *Arduino* além de soluções práticas, confiáveis e de baixo custo para a automação de residências. Para comprovar a eficiência dos dispositivos foi realizado experimento em bancada, onde foi possível monitorar e controlar os circuitos de testes, além de comparar os valores com um multímetro a parte. Perante este experimento, os dispositivos utilizados se comportaram de maneira satisfatória e atenderam aos objetivos propostos.

Os avanços na eletrônica e na tecnologia da computação podem ser facilmente integrados. Conforme descrito neste artigo, existem diversas tecnologias e recursos de baixo custo que podem viabilizar a execução de projetos, trazendo maior comodidade e segurança aos usuários. Também deve ser destacado que o rápido e significativo aumento da demanda pelo uso de dispositivos móveis como *tablets* e *smartphones* é um aspecto positivo, que impulsiona o desenvolvimento de aplicativos que podem afetar o dia a dia dos usuários. Depois de analisar os experimentos, coletar as respostas e avaliar os resultados obtidos, foi possível concluir que a metodologia utilizando o *Node-RED* agregado ao *Home Assistant* na plataforma *Raspberry Pi* promovendo a interação com o usuário e o *Arduino* com sensores e reles na parte de controle permitiu a obtenção de resultados satisfatórios, alcançando, desta forma, os objetivos deste trabalho.

Dentre as principais contribuições trazidas por esta pesquisa, pode-se destacar que o DCR não só apresenta uma automação residencial utilizando recursos livres e acessíveis, como também traz perspectivas sobre o desenvolvimento tecnológicos e como estes podem ser implementados no

cotidiano de qualquer pessoa que venha ser adepta as inovações das redes inteligentes. Tendo isso em vista, é válido dizer que este trabalho apresenta a capacidade que as equipes de desenvolvimento tecnológico possuem para solucionar inúmeros problemas, dos mais simples até os mais complexos.

Para trabalhos futuros, sugere-se a abertura do endereço através de um servidor DNS gratuito, para que o acesso de redes externas possa ser feito sem a necessidade de pagamento de taxas mensais. Tal operação foi impossibilitada no caso deste trabalho devido à dificuldade imposta pelas operadoras de *internet* que bloqueia tal acesso nas configurações dos modems residenciais. Também pode-se sugerir a utilização de dispositivos ainda mais simples com menores custos, que atendam aos requisitos de processamento e consigam suprir as demandas do sistema (por exemplo, a utilização de um *Arduino Uno* no lugar de um *Arduino Mega*).

#### REFERÊNCIAS

- [1] BRAGA, L. C. et al. Estudos Preliminares Para Projeto De Automação Predial Com Eficiência Energética. Monografia (Pós-graduação em Engenharia Elétrica), Universidade Federal de Minas Gerais, Belo Horizonte, 2007. Disponível em: [www.ppgee.ufmg.br/documentos/PublicacoesDefesas/713/LauraBraga\\_CBEE2007.pdf](http://www.ppgee.ufmg.br/documentos/PublicacoesDefesas/713/LauraBraga_CBEE2007.pdf). Acesso em: 01 de setembro de 2020.
- [2] ROQUE, A. Introdução a domótica. In *Revista O Electricista*, nº1 Jul, Agos e Set de 2002. Disponível em: <http://www.antonioroque.com/textos.asp?idCat=11&idArtigo=12>. Acesso em 30 de setembro de 2020.
- [3] GUIMARÃES FILHO, H. T.; SOUZA, R. V. Influência da domótica no mercado imobiliário: estudo de caso. Monografia (Graduação em Engenharia Elétrica), Universidade do Estado de Minas Gerais, Minas Gerais, 2014. Disponível em: [http://www.waltonmartins.com.br/tcc\\_2014\\_Heli\\_RafaelVilela.pdf](http://www.waltonmartins.com.br/tcc_2014_Heli_RafaelVilela.pdf). Acessado em 02/10/2020
- [4] YUSUPOV, R. M.; RONZHIN, A. L. "From Smart Devices to Smart Space" in *Herald of the Russian Academy of Sciences*, v.80, n.1, p.63-68. Rússia: Pleiades Publishing, 2010.
- [5] Weiser, Mark (1991). "The Computer for the 21st Century". In *Scientific American Ubicomp*", vol. 265, no. 3, pp. 66–75.
- [6] RIVERA, R.; ESPOSITO, A. S.; TEIXEIRA, I. Redes elétricas inteligentes (Smart grid): oportunidade para adensamento produtivo e tecnológico local. In *BNDES*, vol. 40, 2013. Disponível em: <http://www.provedor.nuca.ie.ufrj.br/eletrobras/estudos/rivera1.pdf>. Acesso em: 30 de julho de 2018.
- [7] KUP, M. T. Estudo Da Medição Inteligente Para Consumidores Residenciais No Brasil. Monografia (Graduação em Engenharia Elétrica), Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2014. Disponível em: <http://monografias.poli.ufrj.br/monografias/monopoli10014242.pdf>. Acesso em: 30 de julho de 2018.
- [8] FREITAS, C. C. S. de et al. Automação residencial: Cenário atual e perspectivas futuras, in *Revista Ciência e Tecnologia*, v. 15, junho 2012.