

# APLICATIVO ANDROID PARA MONITORAMENTO REMOTO DE MÓDULOS FOTOVOLTAICOS

**Eduardo Silva Piombini** – edupiombini@gmail.com

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Departamento de Engenharia de Controle e Automação

**Wellington Wallace Miguel Melo** – wellingtonwallace@gmail.com

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Departamento de Disciplinas Básicas

**Guilherme Amaral do Prado Campos** – gcampos.cefet@gmail.com

Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, Departamento de Engenharia Mecânica

**Resumo.** *O trabalho proposto consiste em desenvolver um aplicativo que possibilite o acompanhamento remoto dos dados relevantes de um Sistema Fotovoltaico Isolado (SFVI), para isso foi utilizado a plataforma livre Arduino com sensores para obtenção e envio dos dados via rede sem fio. O SFVI monitorado serve para carregar dispositivos móveis (por exemplo, celular e tablets) e é composto por: um módulo FV de 10 Wp, um controlador de carga, bateria e regulador de tensão. Já o sistema de aquisição/monitoramento é formado pelo Arduino ATmega 2560, sensor de corrente, sensor de tensão, sensores de temperatura e de espectro ultravioleta. O aplicativo foi desenvolvido no sistema operacional Android que se mostrou operacional em todos os testes feitos para obtenção dos dados em tempo real.*

**Palavras-chave:** *Sistema Fotovoltaico Isolado, Aplicativo Android, Monitoramento de Dados.*

## 1. INTRODUÇÃO

O Brasil possui um grande potencial de produção de energia elétrica a partir da geração de sistemas fotovoltaicos. Numa comparação entre Brasil e Alemanha, é possível afirmar que o local menos ensolarado do território nacional pode gerar mais eletricidade do que no local mais ensolarado da Alemanha (Pereira et al, 2017). Para ampliar o uso desse tipo de tecnologia, o Brasil aprovou, em 2012, a resolução normativa nº 482 da ANEEL que estimulou positivamente o uso desse tipo de fonte. Além disso, outro fator que impulsionou foi o programa de leilões direcionados a geração solar fotovoltaica de grande escala (Hickel et al, 2016).

Para que haja o aprimoramento dos Sistemas Fotovoltaicos (FV) é necessário acompanhá-los. Isso porque existem diversas variáveis que podem aumentar ou diminuir a produção de energia elétrica, como por exemplo: incidência solar direta e difusa, temperatura dos módulos, sombreamento e sujeira. O monitoramento feito de forma permanente possibilita traçar as curvas I-V que é a principal característica dos módulos FV ou arranjos de módulos FV sob diferentes condições atmosféricas (Phelps et al, 2004; Filho e Coelho, 2016).

Existem diversos sistemas de monitoramento no mercado que permitem a aquisição de dados (Coelho, 2016). Para o caso de sistemas FV conectados à rede pode ser utilizado o próprio inversor cc/ca, visto que a maioria possui função de monitoração e aquisição de dados (Pinho e Galdino, 2014). Para o caso de sistema FV isolado precisa-se de um registrador de dados (*dataloggers*) capaz de armazenar e coletar os valores dos sensores.

O Sistema FV instalado no CEFET, no Campus de Nova Iguaçu, é do tipo isolado, ou seja, ele funciona independente da rede elétrica local. O projeto foi concebido para permitir que docentes e discentes tenham acesso a pontos para carregar os seus celulares utilizando energia solar. Como trata-se de um protótipo que se pretende expandir, é necessário o acompanhamento para saber o potencial de produção de energia elétrica no local. Assim, foi desenvolvido um sistema de aquisição de dados e monitoramento que consiga atender o projeto. Para o sistema de aquisição de dados foi utilizada a plataforma livre Arduino e sensores para obtenção de corrente, tensão, temperatura e espectro ultravioleta (UV).

Como sistema de monitoramento, foi desenvolvido neste trabalho um aplicativo feito com o sistema operacional Android cuja função é mostrar em tempo real os valores das variáveis: corrente, tensão, potência gerada, temperatura e UV. Essa monitoração permitiu o acesso direto aos dados do Sistema FV em qualquer lugar do mundo por fazer conexão direta via rede sem fio, o que facilita analisar os dados mesmos fora do local de operação.

## 2. MATERIAIS E MÉTODOS

O trabalho proposto tem como objetivo principal monitorar remotamente um sistema fotovoltaico isolado cuja função é carregar dispositivos móveis. Para isso, foi concebido um fluxograma do sistema para planejar como será feita a interseção dos diversos dispositivos, conforme ilustrado na Fig. 1.

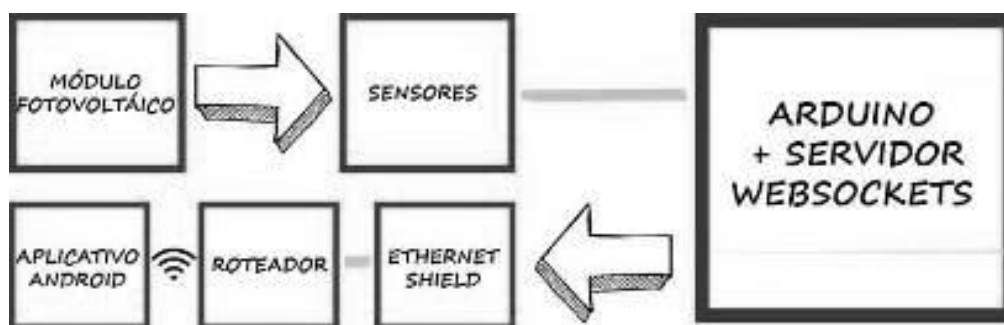


Figura 1 - Fluxograma do Sistema FV.

Para um entendimento do sistema nas próximas seções serão discutidos os componentes que fazem parte do projeto, sendo eles divididos em duas partes: Sistema Fotovoltaico Isolado (SFVI) e do Sistema de Aquisição de Dados / Monitoramento.

## 2.1 Componentes do Sistema Fotovoltaico Isolado (SFVI)

Os Sistemas Fotovoltaicos Isolados (SFVI) são divididos em: individuais ou em miniredes. No individual o projeto atende apenas um único consumidor, enquanto para o caso de miniredes, a produção de energia elétrica pode ser compartilhada por mais consumidores (Pinho e Galdino, 2014). Em ambos, é necessário algum tipo de armazenamento que pode ser feito através de baterias. Elas são incorporadas ao projeto quando se deseja utilizar aparelhos elétricos nos períodos em que não há geração fotovoltaica (Pinho e Galdino, 2014). Por tanto, o projeto do carregador de celular é enquadrado como um sistema individual que utiliza uma bateria para permitir o acesso na parte noturna.

O projeto do carregador de celular é composto pelos seguintes itens: módulo fotovoltaico, controlador de carga, bateria e regulador de tensão, conforme ilustra a Fig. 2.

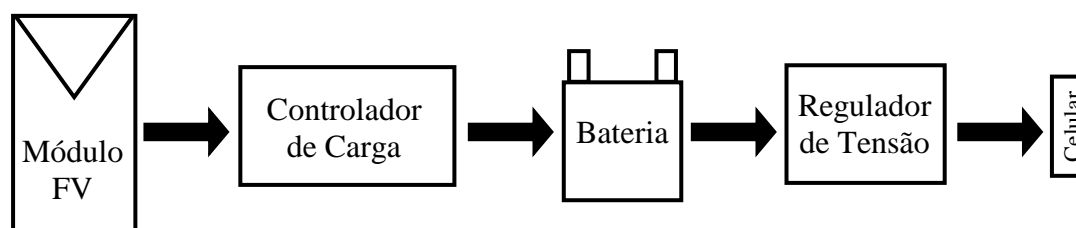


Figura 2 - Esquema do carregador de celular FV.

O módulo FV é o modelo Kyocera KS10 composto por 36 células de silício policristalino e de potência igual à 10 Wp nas condições padrão de teste, 1000 W/m<sup>2</sup> a 25°C e AM 1,5. A tensão de operação é de 17,4 V e a corrente de operação é de 0,58 A. Ele foi instalado na inclinação (*Tilt*) de 22° (semelhante ao valor da latitude local que é de 22.70466°) e rotacionado em relação ao Norte em 20.6° (Azimute igual à 20.6°, Noroeste) para ficar paralelo ao prédio, visto que a perda na geração, relativo a irradiação solar, é de no máximo 2% como ilustra a Tab. 1.

O controlador de carga foi desenvolvido na instituição e possui a característica de transformar a tensão fornecida pelo módulo para que é a tensão de entrada da bateria escolhida. O sistema de armazenamento escolhido é o Haze Power, modelo HMA (6 V / 4,5 Ah) e permite uma autonomia para que o sistema seja utilizado na parte noturna da instituição. Por fim, o regulador de tensão que permite fornecer para o celular a tensão de 5 V que é a característica desse dispositivo, também desenvolvido internamente. Na Fig. 3 é possível observar as instalações do carregador FV.

Tabela 1 - Ábaco de irradiação solar, referente ao CEFET Campus Nova Iguaçu, baseado nos dados do Atlas Brasileiro de Energia Solar.

Inclinação	Ábaco de irradiação Solar [%]						
90°	53%	58%	58%	56%	58%	58%	53%
75°	65%	71%	73%	72%	72%	71%	64%
60°	75%	82%	85%	86%	85%	81%	74%
45°	83%	90%	94%	95%	94%	89%	82%
30°	88%	95%	98%	100%	98%	94%	88%
15°	92%	96%	98%	99%	98%	96%	92%
0°	93%	93%	93%	93%	93%	93%	93%
	90°	60°	30°	0°	30°	60°	90°
	Desvio Azimutal						

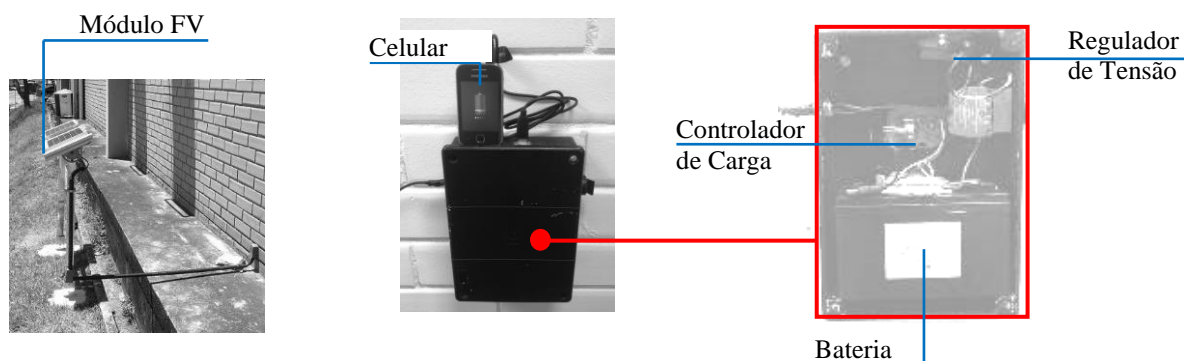


Figura 3 - Instalações do carregador FV.

## 2.2 Componentes do Sistema de Aquisição de Dados / Monitoramento

O sistema de aquisição de dados e de monitoramento foi desenvolvido com o foco de se utilizar tecnologias acessíveis de baixo custo, se comparados com os sistemas de supervisórios e *dataloggers* disponíveis no mercado. Para tornar viável o sistema foi escolhido a plataforma de desenvolvimento Arduino ATmega 2560 que é um microcontrolador (Arduino, 2017), com diversos periféricos complementares que facilitam a programação, na Fig. 4 é possível analisar algumas características do mesmo. Este tipo de microprocessador consiste em uma plataforma de prototipagem open-source, sendo o software multiplataforma flexível e de fácil utilização.



Figura 4 - Placa Arduino Mega 2560.

Assim como outras placas Arduino, esta não possui recursos de rede. Por isso, foi acoplado uma placa de rede Ethernet (Arduino, 2017), como é ilustrado na Fig. 5, para facilitar a comunicação e possibilitar o acesso remoto, via Wi-Fi. Além disso, possui entrada para cartão micro SD para a aquisição de dados.



Figura 5 - Placa Arduino Mega acoplada ao Ethernet Shield.

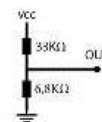
O sensor de Temperatura servirá para medir a temperatura do módulo FV em operação. Ele é um termistor do tipo NTC (Negative Temperature Coeficient) com resistência de 10 k $\Omega$  e impermeável, com precisão de medida de 0,1  $^{\circ}\text{C}$ . Na Fig. 6 é possível observar as características desse sensor.



Figura 6 - Sensor Probe NTC 10K.



Figura 7 - Sensor de tensão P25 GBK.



Para medir a tensão foi escolhido o sensor modelo P25 GBK que é um divisor de tensão com dois resistores de valores 33 k $\Omega$  e 6,8 k $\Omega$  como ilustra a Fig. 7.

Para coletar dados do espectro ultravioleta foi escolhido o sensor de radiação ultravioleta, modelo ML8511, conforme é observado na Fig. 8. Sabe-se que o módulo fotovoltaico de silício policristalino tem pouca influência do espectro ultravioleta para produção de energia, mas foi decidido medir seu efeito sobre o painel para verificar se a

estrutura será afeta ao longo do tempo. Ele monitora o espectro nos comprimentos de onda entre os 280 nm e os 390 nm, faixa que inclui os raios UVA e UVB.



Figura 8 - Sensor ML8511.

O sensor de corrente utilizado para medir é o modelo ACS712 e possui a capacidade de medição igual a 5 A. Esse instrumento usa o efeito hall para detectar o campo magnético devido a passagem da corrente, gerando na saída do módulo (pino OUT), uma tensão proporcional de 185 mV/A (Allegro Microsystems, 2017). Na Fig.9 é visto as características físicas do sensor.



Figura 9 - Sensor de corrente ACS712 5A.

Para validar o microcontrolador (Arduino) e o sensor de tensão (modelo P25 GBK) serão tomadas vinte medidas em três horários ao longo de um dia com céu limpo, ou seja, cerca de 60 medidas. No mesmo período de tempo serão realizadas as mesmas medições com o multímetro digital da marca FLUKE (modelo 600 V CAT III *digital multimeter* 106 e exatidão básica de 0,5%).

### 3. SISTEMA DE AQUISIÇÃO (SERVIDOR) E APLICATIVO

Na abordagem desse projeto, o Arduino atua como um servidor *web* e envia os valores lidos em cada sensor para todos os clientes conectados, conforme ilustra a Fig.10.

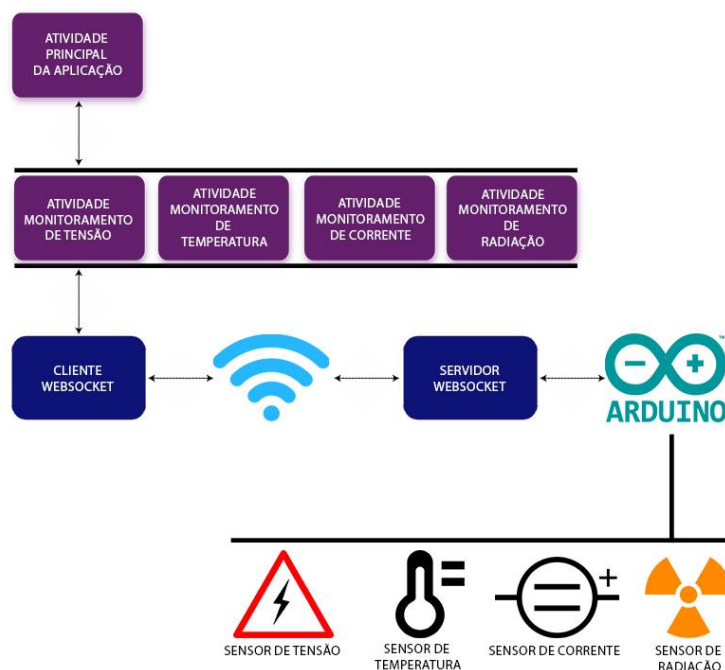


Figura 10 - Fluxograma do sistema de aquisição de dados / monitoramento.

**Código do Servidor.** A comunicação entre cliente-servidor é feita com o protocolo *websockets* por permitir uma transmissão bidirecional entre cliente e servidor. Assim, o cliente não precisa perguntar periodicamente ao servidor se há um novo dado. Ele será notificado assim que eles estiverem disponíveis. A conexão do Arduino à rede é feita através de uma placa Ethernet conectada a um roteador *Wi-Fi*. A configuração da placa Ethernet é bastante simples, basta adicionar à função setup do Arduino as linhas:

```
byte mac_address[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
IPAddress ip(192, 168, 0, 90);
Ethernet.begin(mac, ip);
```

Uma vez configurada a placa de rede, o próximo passo é inicializar o servidor *websockets*. No código proposto foi utilizado a biblioteca *ArduinoWebSocketServer* que pode ser baixada sem custos na referência Arduino WebSocket Server . O primeiro passo em seu uso é incluir seu *header* e criar uma instância de sua classe:

```
#include <WebSocket.h>
WebSocketServer wsServer("", 80, 4);
```

O construtor da classe tem como argumentos: o endereço do servidor, a porta e o número máximo de clientes conectados respectivamente. Como há limitação nos recursos do hardware do Arduino, este número de clientes deve ser escolhido com cuidado. Uma vez criada uma variável global com a instância da classe é feita sua inicialização dentro da função *setup* do Arduino:

```
wsServer.registerConnectCallback(&onConnect);
wsServer.registerDataCallback(&onData);
wsServer.registerDisconnectCallback(&onDisconnect);
wsServer.begin();
```

As linhas acima servem para registrar as funções que serão invocadas em eventos importantes como a conexão de um novo cliente, *onConnect*, ou de dados enviados pelo cliente, *onData*. Sempre que o aplicativo enviar uma mensagem para o Arduino a função *onData* será invocada e dentro dela devem ser tomadas as decisões pertinentes.

Uma vez finalizadas todas as inicializações, foi iniciado a fase de leitura dos sensores e o envio dos valores obtidos para todos os clientes conectados. Isto é realizado na função *loop* do Arduino. É possível observar nas próximas linhas uma versão resumida do código para leitura do sensor de temperatura:

```
void loop(){
    wsServer.listen();
    delay(1000);
    double temperatura = ler_termometro();
    String temperatura_str("T:" + String(temperatura));
    if (wsServer.connectionCount() > 0) {
        wsServer.send(temperatura_str.c_str(), temperatura_str.length());
    }
}
```

Ao enviar os valores para os clientes um caractere de identificação é adicionado no início da mensagem. Por exemplo, uma temperatura de 30°C é enviada na forma “T:30”. Desta forma, identifica-se no código do Aplicativo qual é o sensor está enviando o valor.

No código acima também pode ser visto que a função *delay* do Arduino é invocada em cada loop com um valor de 1 segundo (ou 1000 ms). Esse valor foi escolhido pois a documentação da biblioteca *websockets* diz para não utilizar atrasos longos.

**Código do Aplicativo.** Como acontece com os demais aplicativos Android, existem uma classe *MainActivity* que herda de *AppCompatActivity* e é responsável por gerenciar todo o aplicativo, incluindo a interface gráfica. Nesta classe foi utilizado o método *onCreate* cuja função é realizar as inicializações necessárias. Para associar *call-backs* aos botões da interface gráfica que acionarão a visualização do valor e gráficos pertinente a cada sensor foi adotado que cada um possui sua própria atividade. Isso significa que não é possível visualizar valores e gráficos de mais de um sensor ao mesmo tempo. Nas próximas linhas de código é possível observar uma versão resumida da classe *MainActivity*, apenas com as linhas pertinentes ao sensor de tensão:

```
public class MainActivity extends AppCompatActivity {
    private Button botao_tensao;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        botao_tensao = (Button) findViewById(R.id.button_tensao);
        botao_tensao.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                Intent intent = new Intent(getApplicationContext(), Tela_Tensao.class);
                startActivity(intent);
            }
        });
    }
}
```

A função `setContentview` define de qual arquivo `xml` a interface gráfica deve ser carregada. A função `findViewById` cria uma instância do `widget` cujo rótulo de identificação foi dado como argumento. A função `setOnClickListener` define qual função será invocada quando o botão for clicado. Finalmente com a função `startActivity` é inicializado a atividade que mostrará na tela do celular o valor do sensor e o gráfico de seu valor em função do tempo.

O código da atividade inicializada para cada sensor é similar ao da `MainActivity`. Nela é inicializado um cliente `websockets` que conectará com o Arduino e também é criada uma instância da classe que realizará o gráfico dos valores dos sensores em função do tempo. O cliente `websockets` foi implementado a partir da biblioteca `Java-WebSocket` que está disponível gratuitamente no endereço <https://github.com/TooTallNate/Java-WebSocket>. O uso dela se resume a herdar da classe `WebSocketClient` e implementar os métodos `onOpen`, `onClose` e `onMessage`. A conexão com o servidor é feita criando uma instância da classe e invocando o método `connect`:

```
client = new Client(new URI( "ws://endereço_do_servidor"));
client.connect();
```

Uma vez realizada a conexão, o método `onMessage` será invocado sempre que o servidor enviar uma mensagem.

#### 4. RESULTADOS E DISCUSSÕES

O estudo apresentado está associado ao monitoramento remoto através de um aplicativo Android para verificar a produção de energia elétrica a partir de um módulo FV. O aplicativo apresenta dados em tempo real das leituras feitas pelos sensores em comunicação com o microcontrolador Arduino Mega. Como já dito anteriormente, o sistema de aquisição permite obter os dados de tensão e corrente gerado pelo sistema fotovoltaico, além do índice de espectro UV (no aplicativo aparece como radiação) e temperatura.

O aplicativo permite visualizar os gráficos de cada variável de forma independente, como é ilustrado na Fig. 11. Isso porque, ele foi desenvolvido para o usuário ter acesso a variável escolhida ao apertar um botão. Ao apertá-lo, o aplicativo mostra em tempo real a medição (vide a Fig. 12) com um intervalo de tempo de 1 s entre pontos. Este valor é determinado pelo intervalo de tempo entre iterações do loop do Arduino e foi assim escolhido pois a biblioteca para comunicação `websockets` não trabalha bem com intervalos muito longos. Este intervalo de tempo também pareceu ser um bom compromisso entre resolução e espaço ocupado pelo arquivo de texto onde os dados são salvos no `sdcard`.

O aplicativo é capaz de mostrar 300 pontos de medição no gráfico por ele criado, até que o primeiro ponto seja descartado e as medições continuem sendo feitas e atualizadas normalmente. O usuário também é capaz de expandir ou contrair a imagem gerada pelo gráfico a fim de melhor visualização e observação de pontos específicos.

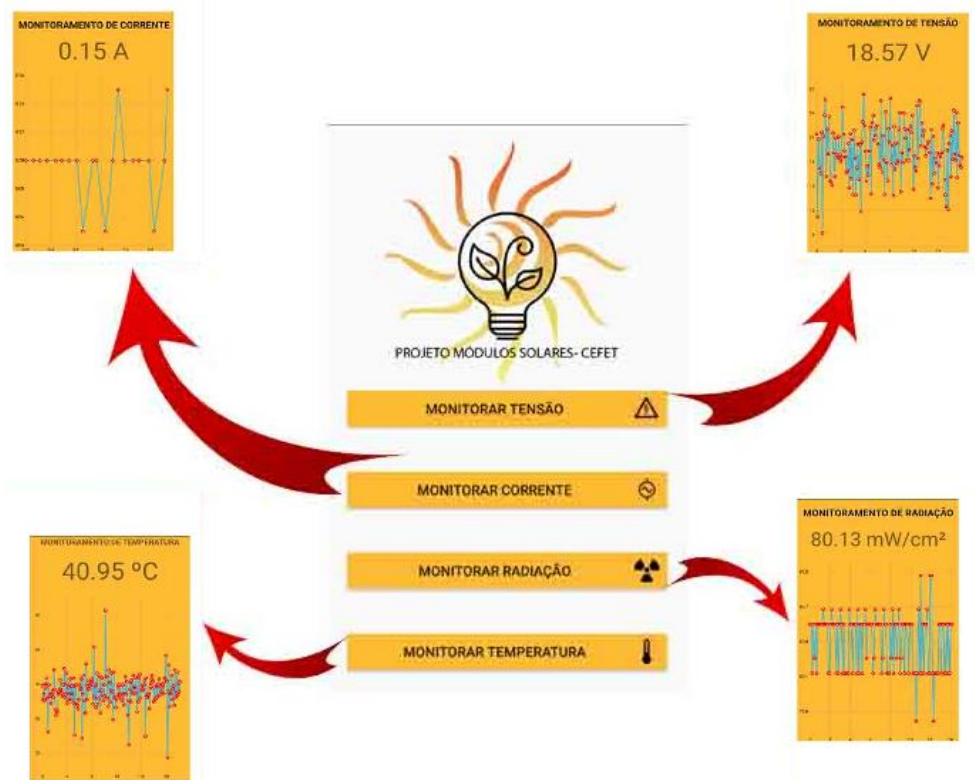


Figura 11 - Tela principal do aplicativo com a ilustração das telas de cada variável.

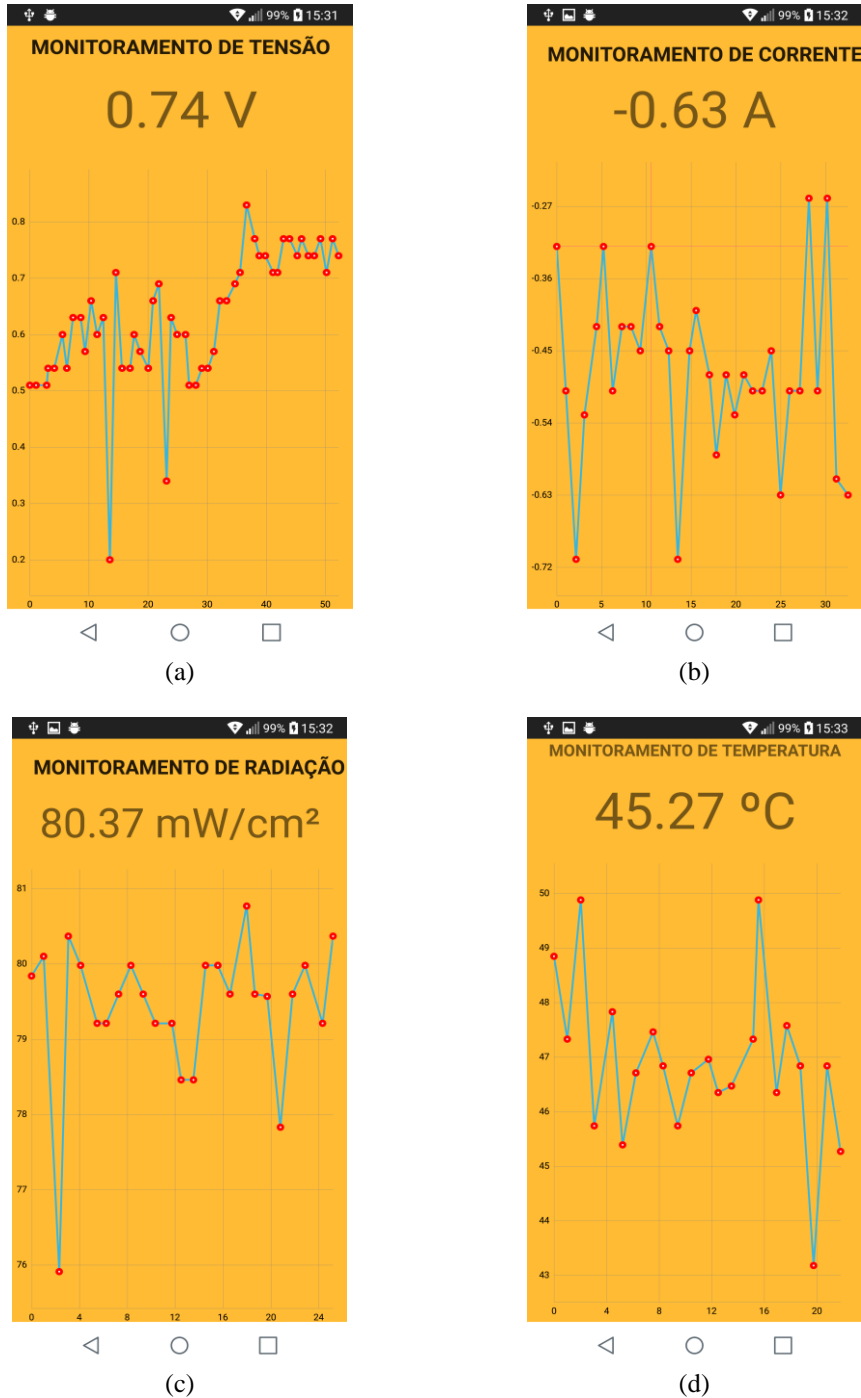
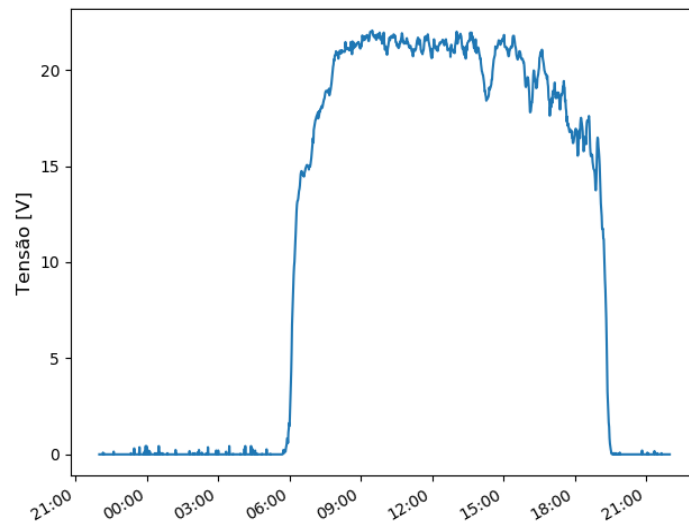


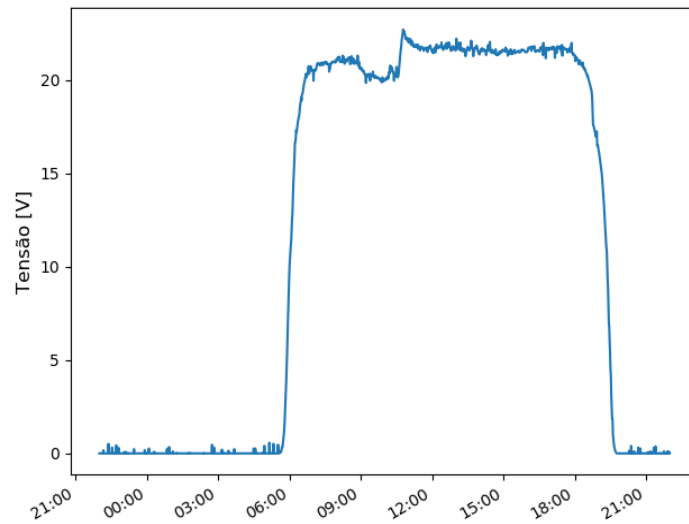
Figura 12 -Telas ilustrativas do aplicativo. (a) Tela do monitoramento de tensão, (b) Tela de monitoramento de corrente, (c) Tela de monitoramento de irradiação UV e (d) Tela de monitoramento de temperatura.

Além do aplicativo, foi acrescentado no projeto a utilização de um cartão de micro SD instalado no *Ethernet Shield* para fazer aquisição de dados ao longo dos dias, para no futuro avaliar o comportamento da geração FV no local ao longo dos anos. Nesse primeiro momento foi realizado a aquisição de dados de três dias com o cartão micro SD e gravado através de arquivos diários em “.txt”. Após a obtenção, os dados foram processados mediante um código desenvolvido utilizando a linguagem de programação *Python*. Na Fig. 13 é possível verificar os valores de tensão ao longo de três dias (11/11, 12/11 e 13/11/2017). Na Fig. 13.a, verifica-se que das 14 horas tem um pico decrescente que pode indicar a passagem de nuvem e após as 16 horas a tensão começa a reduzir devido a densidade de nuvens ter aumentado. Na Fig.13.b, verifica-se um pico crescente aproximadamente as 10:30 horas e depois o gráfico fica constante até as 18 horas, isso ocorre porque o módulo FV pega sombra na parte da manhã nessa época do ano. Já na Fig.13.c, verifica-se uma tensão mais baixa até as 10 horas, depois há um crescimento suave até as 18 horas, isso ocorre

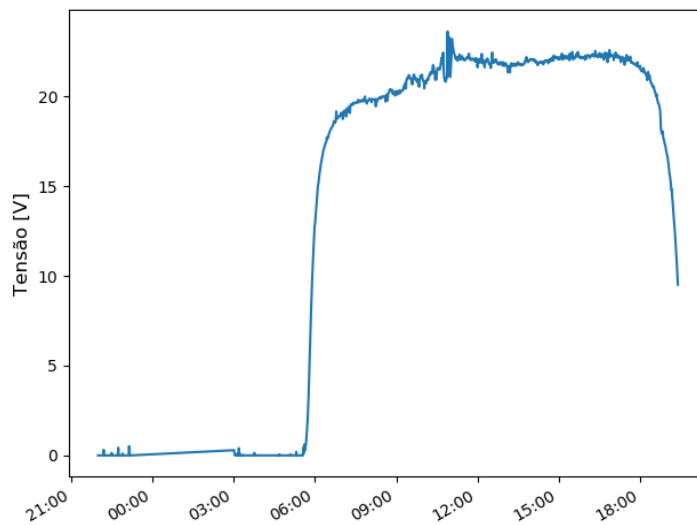
porque o módulo FV pega sombra na parte da manhã nessa época do ano e nuvens que foram se dissipando na parte da manhã. Além da tensão, foram também obtidos os valores de corrente, temperatura e UV para avaliação futura.



(a)



(b)



(c)

Figura 13 - Gráficos de tensão obtidos a partir de dados coletados e processados. (a) dia 11/11/2017; (b) dia 12/11/2017; e (c) dia 13/11/2017.



Para validar os dados de tensão foram feitos um total de sessenta medições (no dia 09/01/2018) com o sensor modelo P25 GBK e o multímetro digital FLUKE modelo 600 V, sendo realizadas vinte medições em três horários distintos: 12 horas, 13 horas e 14 horas. Com o céu limpo, foi possível obter uma incerteza máxima de medição de  $\pm 0,1$  V, conforme é apresentado na Tab. 2.

Tabela 2 - Aquisição de dados com o multímetro modelo 600 V e sensor de tensão modelo P25 GBK.

<b>Medição 1 (12:00)</b>		
Multímetro (V)	Sensor (V)	Desvio Padrão do Sensor
Media	Media	
19,57	19,56	0,07
Incerteza de Medição (V)		
0,1		
<b>Medição 2 (13:00)</b>		
Multímetro (V)	Sensor (V)	Desvio Padrão do Sensor
Media	Media	
19,63	19,63	0,02
Incerteza de Medição (V)		
0,1		
<b>Medição 3 (14:00)</b>		
Multímetro (V)	Sensor (V)	Desvio Padrão do Sensor
Media	Media	
19,54	19,54	0,03
Incerteza de Medição (V)		
0,1		

O custo de materiais do sistema de monitoramento e aquisição de dados é de R\$ 156,80, como é descrito na Tab. 3. Um custo bastante reduzido se comparado com os sistemas disponíveis no mercado.

Tabela 3 - Custo do sistema de aquisição de dados / monitoramento.

Componente	Especificação	Qtd	Valor unit.	Valor Total
Diodo	1N4007	1	R\$ 0,30	R\$ 0,30
Arduino Mega R3	2060	1	R\$ 60,00	R\$ 60,00
Arduino Ethernet Shield	W5100	1	R\$ 30,00	R\$ 30,00
Sonda NTC	10K $\Omega$	1	R\$ 10,00	R\$ 10,00
Sensor de Tensão	P25 GBK	1	R\$ 15,00	R\$ 15,00
Sensor de irradiância UVA	ML8511	1	R\$ 30,00	R\$ 30,00
Sensor de Corrente	ACS712 (5 A)	1	R\$ 10,00	R\$ 10,00
Resistor	1 / 5 W	1	R\$ 1,50	R\$ 1,50
			<b>Valor Total</b>	<b>R\$ 156,80</b>

## 5. CONSIDERAÇÕES FINAIS

Os Sistemas Fotovoltaicos Isolados (SFVI) permitem levar energia elétrica para locais de difícil acesso ou que não dispõem de rede elétrica de qualidade. Nesse trabalho a proposta foi desenvolver um sistema de aquisição de dados e monitoramento para um SFVI que serve para carregar os dispositivos móveis. A concepção do sistema de aquisição e monitoramento utilizou Arduino e Android, assim como sensores de: tensão, corrente, temperatura e espectro ultravioleta (UV). Contudo, esse desenvolvimento necessitou de um estudo profundo sobre os sistemas operacionais e seus componentes para que o monitoramento remoto fosse feito com qualidade.

O aplicativo se mostrou confiável e estável pois ele respondeu aos comandos toda vez que se tentou acessá-lo. Ele possui um acesso imediato da variável desejada apenas acionando o botão da mesma, o que facilita ao usuário a visualização quando o mesmo estiver longe do local de instalação.

O projeto também proporcionou a aquisição de dados com o aplicativo e o cartão micro SD. Com ele foi possível adquirir de forma constante os valores de três dias seguidos e avaliação que o sistema de aquisição e monitoramento está funcionando de forma satisfatória e com uma incerteza de medição de tensão de  $\pm 0,1$  V.

Como projeto futuro pretende-se expandir o aplicativo para os outros módulos FV instalados na instituição e fazer a aquisição de dados ao longo dos anos para verificar o potencial de produção de energia elétrica a partir de sistemas FV. Além disso, utilizar outros instrumentos de medições calibrados para validar os demais sensores de corrente, temperatura e UV.

## REFERÊNCIAS

- Allegro Microsystems, LLC. Fully Integrated, Hall Effect-Based Linear Current Sensor IC. Disponível em: <<https://www.allegromicro.com/~media/Files/Datasheets/ACS712-Datasheet.ashx>>. Acesso: 14 ago. 2017.
- Android. Disponível em: <<https://developer.android.com/guide/components/intents-filters.html>>. Acesso em: 30 maio. 2017.
- Arduino. Arduino Mega 2560. Disponível em: <<https://arduino-info.wikispaces.com/MegaQuickRef>>. Acesso em: 10 jun. 2017.
- Arduino Websocket Server. Disponível em: <<https://github.com/ejeklint/ArduinoWebsocketServer>>. Acesso em: 10 jun. 2017.
- Filho, J. I. de O., Coelho, W. M., 2016. Sistema de Aquisição para Painéis Fotovoltaicos com Armazenamento de Dados em Servidor Remoto Utilizando Plataformas Open Source Raspberry PI e Arduino, VI CBENS – VI Congresso Brasileiro de Energia Solar, Belo Horizonte.
- Hickel B. M., Deschamps, E. M., Nascimento, L. R., Rütther, R., Simões, G. C., 2016. Análise da Influência do Acúmulo de Sujeira sobre Diferentes Tecnologias de Módulos FV: Revisão e Medições de Campo, VI CBENS – VI Congresso Brasileiro de Energia Solar, Belo Horizonte.
- Pereira, E. B., Martins, F. R., Gonçalves, A. R., Costa, R. S., Lima, F. J. L., Rütther, R., Abreu, S. L., Tiepolo, G. M., Pereira, S. V., Souza, J. G., 2017. Atlas Brasileiro de Energia Solar. INPE, 2 ed, São José dos Campos.
- Pinho, J. T., Galdino, M. A., 2014. Manual de Engenharia para Sistemas Fotovoltaicos. CEPREL – DTE – CRESESB, Rio de Janeiro.
- Phelps, R., Lo, B., Michael, S., 2004. “Evaluation and testing of the solar cell measurement system onboard the naval postgraduate school satellite NPSAT1” in Proc. 22nd AIAA Int. Commun. Satellite Syst. Conf. Exhibit (ICSSC), Monterey, CA, May 9–12, pp. 1–8.
- Websockets. Disponível em: [https://www.tutorialspoint.com/websockets/websockets\\_overview.htm](https://www.tutorialspoint.com/websockets/websockets_overview.htm)>. Acesso em: 10 jun. 2017.

## ANDROID APPLICATION FOR REMOTE MONITORING MODULES PHOTOVOLTAIC

**Abstract.** *The purpose of the work consists in developing an application that allows the remote monitoring of the relevant data of an Photovoltaic System Isolated (PVSIs), for which the Arduino free platform was used with sensors to obtain and send data through wireless network. The monitored PVSIs serves to charge mobile devices (eg, cell phones and tablets) and consists of: a 10 Wp PV module, a charge controller, battery and voltage regulator. Already the acquisition / monitoring system is formed by the Arduino ATmega 2560, current sensor, voltage sensor, temperature sensors and ultraviolet spectrum. The application was developed in the Android operating system that proved to be operational in all tests made to obtain the data in real time.*

**Key words:** *Isolated Photovoltaic System, Android App, Data Monitoring.*