https://eventos.utfpr.edu.br//sicite/sicite2020

Desenvolvimento de sistema web para monitoramento de dados de usinas solares

Development of a web system for monitoring data from solar power plants

João Pedro da Matta Galera da Silva
joaosilva.2016@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Jorge Assade Leludak
assade@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Curitiba, Paraná, Brasil

Recebido: 19 ago. 2020.
Aprovado: 01 out. 2020.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

RESUMO

Seguindo a implantação das Estações de Pesquisa em Energia Solar, o trabalho buscou implementar uma interface para monitoramento de tais estações. Foi desenvolvida, então, um sistema web utilizando a linguagem de programação Python e o *framework* Django. Esta solução utiliza os dados brutos enviados pelas estações a um servidor FTP e os exibe em uma página acessível por qualquer dispositivo conectado à internet que possua um navegador web, possibilitando o monitoramento em tempo real, pelos pesquisadores, de diversas variáveis coletadas nas estações.

PALAVRAS-CHAVE: Telemetria. Aplicações web. Armazenamento de dados. Python (linguagem de programação de computador).

ABSTRACT

Following the implantation of the Solar Energy Research Stations, the work looked for implement a interface for monitoring those stations. It was developed a web system, using the Pyhton programming language and the Django framework. This solution uses the raw data sent by the stations and shows then in a page acessible by any internet connected device whtch has a web browser, allowing the real time monitoring, by the researchers, of several variables collected on the stations.

KEYWORDS: Telemetry. Web applications. Data warehousing. Python (computer program language).

INTRODUÇÃO

A rede de estações de pesquisa em energia solar, implantada em um projeto de pesquisa desenvolvido pela UTFPR e pela Copel, abrange seis estações localizadas em seis campi da UTFPR. Em cada estação existem estações solarimétricas e ambientais que registram dados meteorológicos e de irradiação solar, além de módulos de avaliação de diferentes tecnologias de painéis fotovoltaicos (SILVEIRA e URBANETZ JR, 2019; REDE...,2019).

Nessas estações, os dados solarimétricos e ambientais são coletados por equipamentos datalogger modelo Campbell Scientific CR1000X, enquanto os dados de geração são coletados dos inversores por um microcontrolador através de uma rede RS485. Todos os dados processados por esses dispositivos são salvos em arquivos do tipo '*comma separated values*' (CSV) e enviados a um servidor através do protocolo '*File Transfer Protocol*' (FTP) (REDE...,2019).

Com essa abordagem, o acesso aos dados envolvia acessar o servidor, baixar os arquivos individualmente e então utilizar alguma ferramenta para analisar as informações. O problema é que, com este método, não se tinha informação em tempo real do que acontecia nas estações. Problemas com equipamentos e inconsistências nos dados, por exemplo, só eram identificados algum tempo depois.

Nesse contexto, foi realizado o presente trabalho, de forma a criar uma solução para visualizar em tempo real os dados coletados pelas plantas solares.

METODOLOGIA

A ideia proposta era montar um painel com monitores onde seriam apresentados os dados. Para isso foram adquiridos seis televisores Samsung UN32J4290AG, cada um para apresentar os dados um campus. Por serem televisores do tipo *Smart*, com acesso à internet e navegador web integrado, optou-se pelo desenvolvimento do sistema web, que poderia ser executado nativamente na TVs, sem a necessidade de um equipamento externo, além de permitir que o monitoramento também pudesse ser realizado de fora do laboratório.

A aplicação é dividida em duas partes: o *front-end*, que se trata da interface que é exibida ao o usuário, incluindo as páginas web em si; a exibição dos gráficos; a adaptação da página a diferentes formatos de tela; entre outros, enquanto o *back-end* trata do processamento realizado sobre os dados que serão exibidos, incluindo a identificação das solicitações dos usuários; o acesso e leitura dos arquivos CSVs; a seleção das variáveis enviadas ao usuário; identificação de possíveis erros; a geração da página; entre outros.

O *front-end* foi desenvolvido, primariamente, em HTML5, que é a evolução mais recente da HyperText Markup Language (HTML). O HTML é o elemento básico da Web e é responsável por estruturar e dar significado semântico ao conteúdo. Em sua quinta versão, trouxe tecnologias que permitem efetivamente o desenvolvimento de poderosas aplicações Web (HTML, 2019; HTML5, 2019).

Em conjunto com o HTML é utilizado o CSS (Cascading Style Sheets). Essa linguagem é a responsável por definir como o conteúdo é exibido na tela, incluindo

disposição de elementos; cores; formatação de texto; tratamento para telas de diferentes tamanhos, etc (CSS, 2019).

Por fim, o *front-end* utiliza o *JavaScript*, uma linguagem de programação orientada a objetos que é interpretada pelo próprio navegador do usuário (JAVASCRIPT, 2019). Nesse sistema de monitoramento, a principal função do *JavaScript* é a renderização dos gráficos a partir dos dados enviados pelo *back-end*. Para essa função, foi escolhida a biblioteca *Charts.js*, uma flexível biblioteca JavaScript de código aberto que permite a geração de gráficos de forma simples (BJERROME, [20-?]).

Já no *back-end* optou-se por utilizar a linguagem Python, uma linguagem de programação de alto nível que, devido à sua simplicidade e versatilidade, somado a um grande número de bibliotecas existentes, permite o desenvolvimento de poderosas aplicações (PYTHON..., 2019), sendo muito utilizada para análise de dados e escolhida para diversas funções neste projeto de pesquisa.

No desenvolvimento, foi utilizado o *framework* Django. Um *framework* é um pacote se *software* que reúne funcionalidades comuns a várias aplicações (O QUE..., [20-?]). No caso do Django, essas funcionalidades se apresentam como uma plataforma para o desenvolvimento web em Python. Este framework é o responsável por receber as solicitações dos usuários e executar o código correspondente, prover a interface para o código tratar a requisição e gerar a página com base em um *template* previamente desenvolvido e nos dados definidos pelo código responsável por aquela requisição (INTRODUÇÃO..., 2020).

Assim, quando um usuário deseja visualizar um determinado conjunto de dados, este acessa a Uniform Resource Locator (URL) da página que apresenta esses dados através de um navegador web. A requisição é enviada ao servidor onde a aplicação está hospedada através do protocolo *HyperText Transfer Protocol over SSL*. No servidor é definido o código a ser executado, que por sua vez define os arquivos CSV e dados a serem utilizados. Após lidos e processados, estes dados são usados na geração da página, que é então enviada ao navegador do usuário, onde é renderizada e as informações enfim apresentadas (UMA..., 2020).

Como os arquivos CSV estão salvos no servidor FTP, é preciso ainda, durante o processamento da requisição, acessar este servidor. Para isso foi escolhido o protocolo SMB (Server Message Block), um protocolo que permite o compartilhamento de recursos entre computadores. Este protocolo permite que um diretório em um dispositivo remoto possa ser montado como uma pasta local (SHARPE, 2002), apresentando um baixo tempo de resposta no acesso aos dados. Assim, o sistema de monitoramento trata os arquivos CSV como se estivessem localizados localmente, com o sistema operacional sendo o responsável pela comunicação com o servidor FTP de forma totalmente transparente.

Além do monitoramento por iniciativa do usuário, foi implementado um sistema automático de notificação de problemas com o recebimento dos dados pelo servidor FTP. Também desenvolvido em Python, esse programa é executado a cada hora no servidor, verificando quanto tempo cada arquivo CSV está sem atualização. Os arquivos são separados em grupos de acordo com o dispositivo responsável por aquele arquivo, sendo um grupo para cada datalogger e um grupo para cada microcontrolador responsável pelos dados dos inversores. Se um grupo

inteiro estiver sem atualização há mais de uma hora, é enviado um *e-mail* de notificação para os usuários previamente cadastrados.

RESULTADOS

O sistema de monitoramento foi disponibilizado na internet através do endereço https://geracao.labens.ct.utfpr.edu.br/painelCampus/. Ao acessa-lo, é fornecido um *menu* que permite escolher qual estação solar se deseja visualizar, identificada pelo campus onde está instalada (Figura 1). Há ainda o menu 'opções avançadas' que permite realizar uma consulta rápida nos dados de dias anteriores e, no caso de Curitiba, que possui dois tipos de estações solarimétricas, SONDA (Sistema de Organização Nacional de Dados Ambientais) e EPE (Empresa de Pesquisa Energética), escolher a estação da qual se deseja obter os dados (Figura 2).

Figura 1 – Menu de seleção do sistema de monitoramento

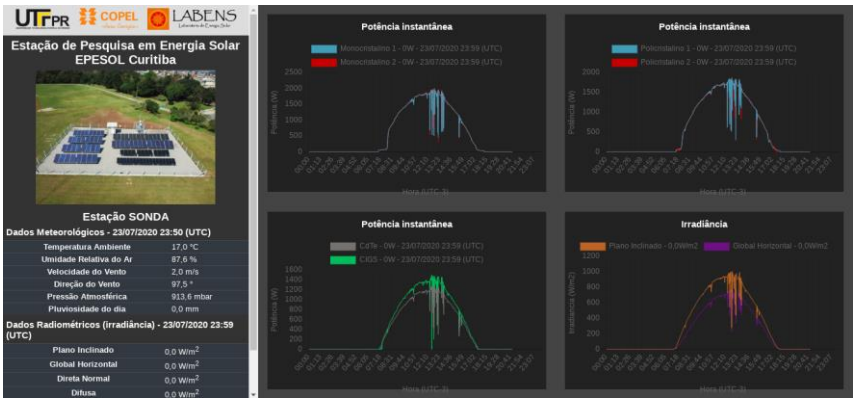
Fonte: Autoria própria.

Figura 2 – Menu de opções avançadas

Fonte: Autoria própria.

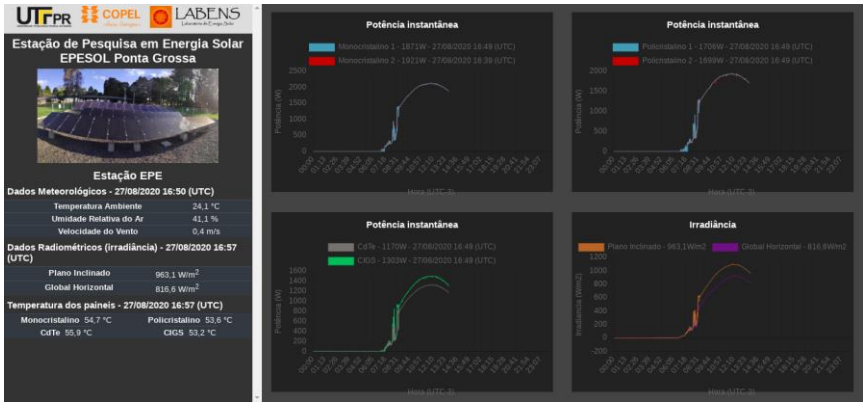
Após a escolha dos dados, é efetivamente mostrado um painel com as informações. Com gráficos de geração de energia e irradiância ao longo do dia, além das informações meteorológicas mais recentes e da temperatura dos painéis solares, a página consegue se adaptar a vários tipos de telas. Na figura 3 é mostrada a interface de uma estação do tipo SONDA em um computador ao fim do dia, enquanto na figura 4, também pelo computador, é mostrada uma estação EPE durante o dia.

Figura 3 – Estação Campus Curitiba ao fim do dia 23/07/2020 em um computador



Fonte: Autoria própria.

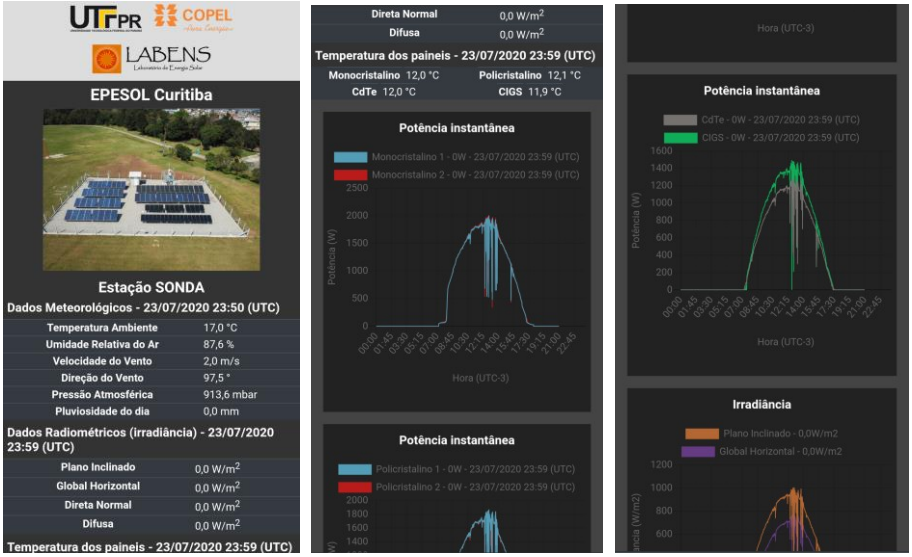
Figura 4 – Estação Campus Ponta Grossa as 14h do dia 27/08/2020 em um computador



Fonte: Autoria própria.

A visualização da página em celulares pode ser observada nas figuras 5, 6 e 7.

Figura 5, 6 e 7 – Estação Curitiba ao fim do dia 23/07/2020 em um celular



Fonte: Autoria própria.

Por fim, para o monitoramento no laboratório, foram instalados os 6 televisores adquiridos e seus navegadores configurados, cada um, para abrir, por padrão, a página de um campus. O resultado é mostrado na figura 8.

Figura 8 – Televisores de monitoramento do laboratório

Fonte: Autoria Própria.

Na figura 8 é possível visualizar alguns gráficos vazios, indicando a existência de problemas com aqueles dispositivos.

Quanto ao envio de *e-mails*, é possível observar na figura 9 em exemplo onde é indicado a falta de dados dos arquivos gerados pela estação solarimétrica SONDA do campus Medianeira. Com esse e-mail, é possível que o responsável por esse equipamento seja notificado de forma rápida da existência do problema.

Figura 9 – *E-mail* notificando sobre problemas com a estação SONDA de Medianeira

LABENS
Laboratório de Energia Solar

Olá.

Foram detectados alguns problemas na comunicação das EPESOLs com o Servidor FTP.

A(s) tabela(s) listada(s) a seguir não é(são) atualizada(s) há mais de 1 hora.

O padrão é que a atualização seja feita a cada, no máximo, 10 minutos.

MD-temp
MD-sons
MD-so01
MD-so10

Mais detalhes no site https://geracao.labens.ct.utfpr.edu.br/envios

Acesse o link acima para desativar o alarme para esse(s) arquivo(s). Novos e-mais serão enviados a cada hora até que o problema seja resolvido ou o alarme seja desligado.

Universidade Tecnológica Federal do Paraná
Campus Curitiba
Laboratório de Energia Solar

Fonte: Autoria própria.

CONCLUSÕES

Com 9 meses de operação, o sistema de monitoramento tem se mostrado crucial para o desenvolvimento das pesquisas. Se por um lado as análises dos dados ainda dependem do *download* individual dos arquivos no servidor FTP, por

outro, o sistema permite um panorama em tempo real do estado das plantas fotovoltaicas. A solução, é claro, não resolve os problemas sozinha. Suas funcionalidades, apesar de muito úteis, dependem de recursos humanos para identificar corretamente a causa da falha, que pode aparecer de diversas formas: Na rede; no *software*, no *hardware*, entre outros.

O benefício que se obteve foi a identificação rápida de possíveis problemas. Com isso, a equipe técnica pode agir assim que a falha é identificada, evitando que, por exemplo, a falta de dados seja percebida apenas quando estes forem necessários pra a continuidade da pesquisa, que acabava sendo atrasada pelo processo de recuperação das informações. Mesmo nas situações em que ocorrem perdas de dados, a quantidade do que foi perdido diminuiu, reduzindo prejuízos aos pesquisadores e ao projeto como um todo.

AGRADECIMENTOS

Os autores agradecem a UTFPR pelo apoio e infraestrutura disponibilizada para o desenvolvimento desta pesquisa e a COPEL-Distribuição pelo apoio e financiamento dos recursos para realização deste projeto de P&D "ANEEL PD 2866-0464/2017 - Metodologia Para Análise, Monitoramento e Gerenciamento da GD por Fontes Incentivadas".

REFERÊNCIAS

BJERROME, Tobias Ahlin. **Data visualization with Chart.js: An introduction**. Tobias Ahlin, [20-?]. Disponível em: https://tobiasahlin.com/blog/introduction-to-chartjs/. Acesso em 14 de agosto de 2020.

CSS. **MDN Web Docs**, 2019. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/CSS. Acesso em 14 de agosto de 2020.

HTML. **MDN Web Docs**, 2019. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/HTML. Acesso em 14 de agosto de 2020.

HTML5. **MDN Web Docs**, 2019. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/HTML/HTML5. Acesso em 14 de agosto de 2020.

INTRODUÇÃO ao Django. **MDN Web Docs**, 2020 Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Django/Introdução. Acesso em 17 de agosto de 2020.

JAVASCRIPT. **MDN Web Docs**, 2019. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/JavaScript. Acesso em 14 de agosto de 2020.

O QUE é um framework? **Computação UFCG**, [20-?]. Disponível em: http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/frame/oque.htm. Acesso em 17 de agosto de 2020.

PYTHON: 10 motivos para aprender a linguagem em 2019. **Computerworld**, 2019. Disponível em: https://computerworld.com.br/2019/09/15/python-10-motivos-para-aprender-a-linguagem-em-2019/. Acesso em 17 de agosto de 2020.

REDE de estações de pesquisa em energia solar. **LABENS**, 2019. Disponível em: https://labens.ct.utfpr.edu.br/sobre/rede-de-estacoes-de-pesquisa-em-energia-solar/. Acesso em 14 de agosto de 2020.

SHARPE, Richard. **What is SMB?** Samba, 2002. Disponível em: https://www.samba.org/cifs/docs/what-is-smb.html. Acesso em 17 de agosto de 2020.

SILVEIRA, C. O., URBANETZ JR, J. **Estimativa de Desempenho de Sistemas Fotovoltaicos Conectados à Rede em Diferentes Regiões no Paraná**. Smart Energy 2019 Paraná, Curitiba, 2019. Disponível em: https://utfpr-ct-static-content.s3.amazonaws.com/labens.ct.utfpr.edu.br/wp-content/uploads/2019/12/SILVEIRA-et-al-Smart-Energy-2019-V4.pdf. Acesso em 14 de agosto de 2020.

UMA visão geral do HTTP. **MDN Web Docs**, 2020. Disponível em: https://developer.mozilla.org/pt-BR/docs/Web/HTTP/Overview. Acesso em 17 de agosto de 2020.