A Web-Based System for Photovoltaic Energy Generation: Use of Mathematical Models for Irradiance in Brazil

Isabelle F. S. Nunes[1], **Tárcio A. S. Barros**[1], **Denis G. Fantinato**[1]

[1] School of Electrical and Computing Engineering (FEEC),
Universidade Estadual de Campinas (UNICAMP)
Av Albert Einstein 400, 13083-852, Campinas, SP, Brazil.

dev.isabellefsnunes@outlook.com, {tarcio87, denisf}@unicamp.br

Abstract. *In the context of renewable energy generation, software tools are essential for addressing the sector's main challenges. For instance, Photovoltaic (PV) energy demands a study of the PV plant allocation before its deployment to improve efficiency. However, the visualization tools that perform this task face some usability issues, such as the complementary artifacts to analyze and data-driven modeling. In that sense, this work proposes an artifact with an intuitive and user-friendly platform built as a web-based application with open-source technologies, considering the PV mathematical models. The web-based system gathers a set of mathematical models available and was tested for Brazilian irradiance simulation, allowing inclined plane analysis and other variants.*

1. Introduction

The energy sector is one of the major sources of CO_2 emissions due to the combustion of fossil fuels. To reduce Greenhouse Gas (GHG) emissions, one of the highlights of the Kyoto Protocol [UNFCCC 1997] is to promote clean energy generation. The most used and well-known modes for generating renewable energy are hydroelectric, PV, wind, thermoelectric and biomass [Villalva 2015]. This work is focused on PV energy, which has made a significant advance in recent years, as it considers the wide potential of solar irradiation available. Particularly, in Brazil, the PV sector achieved a significant leap of 79.8% between 2021 and 2022, equivalent to 11 GWh. Consequently, for a solid improvement in the sector, challenges involve seeking higher performance and best practices regarding infrastructure to obtain energy return.

In light of this, the study performed by [Lima et al. 2020] emphasizes the importance of simulators in the design of a PV plant by considering situations of greater complexity, such as system losses, ranging from losses due to shading to losses due to the sizing of the infrastructure cables. Such information, when taken into account, increases the accuracy of the estimated result for measuring the efficiency of a PV plant before its deployment and, consequently, decreases the loss costs.

Although there are a few software tools that aim to address these topics, they usually present additional requirements, such as the installation of complementary systems or extension codes, sometimes compromising the user experience, portability, and maintainability. In that sense, the objective of this work is to present an intuitive, user-friendly platform drawing on the general results of simulations. To execute this, the chosen approach involved building a web application with open-source components to simulate the

PV system from mathematical models. For analysis, irradiance simulation in Brazil is considered.

This work is organized in Section 2 to present Photovoltaic concepts to insert the reader into a comprehensive result analysis; Section 3 content shows the related works. Meanwhile, Section 4 with the ideas, and current technologies applied, and Section 5 describes the development with the description of the main components of the layout page in their functionalities. Section 6, concludes the work purpose.

2. Photovoltaic fundamentals

In the context of solar energy, radiation is the main source, i.e. solar rays emit heat and electromagnetic waves that can be transferred to kinetic energy through the Photovoltaic Effect with the specific surface [Villalva 2015]. To characterize the instantaneous radiation and the power received in a period, they are respectively Irradiance (W/m^2) and Irradiation (Wh/m^2). Note that both consider the area of reception, however, rays are subject to trajectory interferences.

The path of irradiances, considering Earth's atmospheric layer and objects of the environment, could decrease the energy that could be obtained, this behavior can exist in direct, diffuse, and total irradiance. In PV systems or databases, these elements are arranged according to the attributes: Global Horizontal Irradiance (GHI) is the irradiance that considers the beam and diffuse components, also known as Direct Normal Irradiance (DNI) and Diffuse Horizontal Irradiance (DHI), respectively.

Considering the collection of irradiances with specific tilts, the Global Tilted Irradiance (GTI) measure can be provided by sensors installed *in situ*. However, the study presented by [Hay and McKAY 1985] emphasizes the methods of calculating with mathematical modeling instead of collecting independently. It uses irradiance values as Plane of Array (POA), DNI and DHI, from data of GHI provided in meteorological or solarimetric databases and additional information from the model, such as tilt angles or constant coefficients.

For detailed PV studies, the meteorological information, infrastructure data, and characteristics of the components to be installed are applied in the mathematical models. For meteorological data, such as irradiance there are public databases with specific coordinates or estimates by satellite integrations, and also, some methods use them as reliable information for simulation.

3. Related works

This paper focuses on web-based applications in public and open research and databases for photovoltaic systems. That way, the relevant sites with the same perspective of diffusion of PV sector advances were listed. This section aims to present the distinct approaches available and emphasize the usability and importance of their available data.

The Laboratory of Modeling and Renewable Energy Resource Studies (LABREN)[1] created the Brazilian Atlas of Solar Energy [INPE and LABREN 2017, Bueno Pereira et al. 2017] offers two types of visualizations, displayed both in purely

[1]The original name is Laboratório de Modelagem e Estudos de Recursos Renováveis de Energia

numerical form and graphically through files with Shapefiles and QGS extensions. It is the most considered data source for this initial model, which provides national monthly and annual irradiance data for each Brazilian city.

SunData 3.0 [Bruno Montezano et al. 2017] also provides irradiance data based on coordinates. In this case, when the coordinate is applied, a search is conducted for the three nearest points, i.e. PV plant associated. It creates plots with irradiance data in both the horizontal and inclined planes at their optimal angles. The initial page layout of SunData has a dated design with simplified structures with technical documentation of attributes, available files and references.

The Global Solar Atlas [Solargis et al. 2019a, Solargis et al. 2019b] is not strictly national but also displays data for Brazil, in addition, it is product-oriented, emphasizing commercial aspects instead of academic perspectives. Upon selecting the coordinates, another page opens with a specific layout for reporting with annual irradiance tables, satellite maps, horizon maps, sunpaths, and DNI at various scales, annual to hourly. It also provides information for residential PV sizing.

These web applications are consolidated in the PV market, however, they do not have release updates, thus implying the usability only for introductory approaches. The purpose of this work will be explained in the next sections. The system has an architecture of web applications that can apply PV mathematical models in a user-friendly layout.

4. Proposal and scope of the system

This work proposes a web-based application to provide intuitive usability independent of the user's technical knowledge. In this instance, the perspective of the web eliminates the need for desktop installations, streamlining the update process for both users and developers. Web-based deployment also reduces hardware requirements on the user's machine, allowing for wider system accessibility and additional features and improvements.

Figure 1 presents the system's main components within the software development field in a concise technical format. This representation illustrates the complete data flow, starting from extracting specific sensors and progressing through various levels of data structuring. The data is shared across synchronized databases, undergoes processing, and is utilized by specialized applications at the visualization layer.

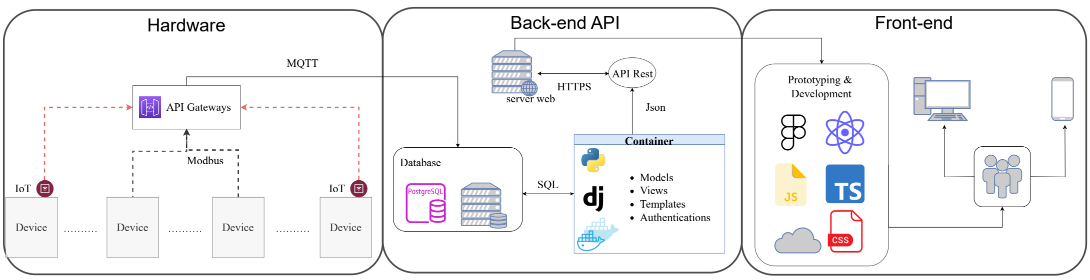


Figure 1. Flowchart of data on Web Application.

The hardware component is dedicated to collecting relevant data to populate the database, including meteorological data, solar irradiance data, and equipment setup data, that can be collected within a photovoltaic plant. These data can be directly extracted by dedicated sensors, such as collecting various types of irradiance data through pyranometers, or connections between Internet of Things (IoT) devices [Kalay et al. 2022] and the plant's infrastructure equipment, such as power absorbed by the photovoltaic cells [Gray 2010, Whitaker et al. 2010]. Intermediate devices like Dataloggers [Kalay et al. 2022] may facilitate the visualization of this data transferred between the collection points and the storage infrastructure, whether physical servers or cloud-based. The available data and platform could be constantly updated with real-time data, which could later be used in research considering Machine Learning (ML) approaches to photovoltaic systems simulations [Gaviria et al. 2022].

Each larger component (Hardware, Back-end, and Front-end) can possess its unique characteristics and specifications, however, they must maintain common data exchange protocols. This application aims to emphasize data visualization functionalities within the photovoltaic sector. In the following subsections, the background of visualization tools will be described.

4.1. Back-end

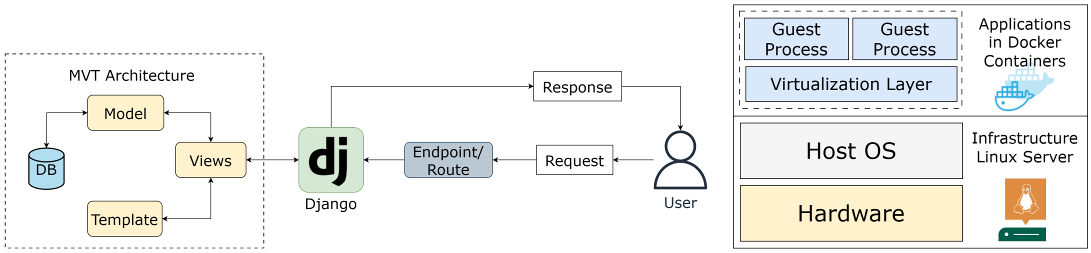
The system component responsible for organizing and processing data is named Back-end. In this stage, the core algorithm governing the system's operation is implemented, encompassing crucial actions, calculations, and analyses through mathematical modeling [da Silva et al. 2018, Klein 1977, de Souza Silva et al. 2022, de Souza Silva et al. 2020].

The code structure responsible for modeling the internal system, enabling the transfer of information from the Hardware to the Front-end, has been developed. Python was chosen as the programming language to standardize this system with other studies involving mathematically simulated stages. Considering the decision to use Python, packages and libraries suitable for web development objectives were sought. The most common options for Python web applications include Django, Pyramid, Bottle, Tornado, Flask, and web2py. However, the choice among these possibilities depends on the chosen approach for implementation, as there are more specialized options, as shown in [Python 2024], that describe relevant characteristics to aid developers in making decisions.

The chosen framework, Django, follows a Model, View, and Template (MVT) architecture, which involves subdividing the continuous implementation stages specific to the system into three cores. The Model focuses on defining each object within the system and establishing connections through migration. Views are dedicated to the behavior and processing of information to create a user-visible response based on the request inputs. Templates are responsible for creating the user interface via HTML and CSS, facilitating the connection between input messages and, consequently, the outputs after specific actions within the Views, appropriately relating to the model.

Figure 2(a) illustrates, through a flowchart, the connection between the MVT architecture and how it is positioned within a web application utilizing the Django framework. It is important to note that Django and its internal configurations serve as the foundation and provide the appropriate means to orchestrate the MVT architecture, making it programmable and internally expandable to enhance project flexibility.

Django features an interface that configures, manages, and provides access routes to each method created by the developer. When a user creates and sends a request to a specific endpoint, Django recognizes this information and processes it accordingly, For



(a) Django Application.

(b) Layers of Container-Based system (based on [Kozhirbayev and Sinnott 2017]).

Figure 2. Back-end characteristics.

instance, it can provide data *as is* from the database or also aggregated data like averaged values. The framework provides predefined responses for common Create, Read, Update, and Delete (CRUD) operations that can be handled without specific application treatment, saving time and resources.

The Django REST Framework have been employed to simplify the development. It extends the object-oriented paradigm, enabling more direct integrations with the chosen Database Management System (DBMS) and optimized serialization. This is achieved using ClassModelView, a framework capable of adjusting elements based on object inheritance, creating code with more connections and no loss of information during implementation.

Evaluations presented in [Fraczek and Plechawska-Wojcik 2017] showcase its high performance for web applications and the [Plechawska-Wójcik and Rykowski 2016] demonstrates PostgreSQL's speed in writing data to storage, particularly for simple attributes and smaller data with few simultaneous bytes. Drawing on the perspective of open-source solutions and object relation database management, PostgreSQL is simple to implement, and results demonstrate that performs well in web application.

Docker is a tool capable of configuring an isolated infrastructure tailored solely for the desired application and its specific functionality. This encapsulation is designed to prevent interference between applications and is known as Container-Based [Soltesz et al. 2007] virtualization. Figure 2(b), based on the image shown in [Kozhirbayev and Sinnott 2017], illustrates the levels of isolation, their respective focuses, and the technology selected for this work. The choice to use Docker as a virtualization platform requires a Unix-dependent infrastructure, and consequently, the server must be pre-configured to accommodate this Operating System (OS).

4.2. Front-end

In this section, the component responsible for direct user interaction, known as the front-end [Godboly 2016], will be introduced. This subdivision can operate independently of the Back-end implementation stage, which combines structure, styles and actions for each functionality, the pillars of stacks are HTML, CSS and JS. Numerous frameworks assist in manipulating these languages and crafting elements more intuitively for developers can focus on achieving the desired design, system responsiveness, and other User Experience (UX) and User Interface (UI) metrics, rather than being overly concerned about the internal code.

At this point, the approach described in [Zammetti 2022] has been chosen as the initial technology, the ReactJS framework was selected for layout implementation, which relies on JS technologies such as Node.js. Additionally, TypeScript is an extension of the JavaScript language that enhances the precision of internal elements and attributes through predefined types, reducing runtime errors, such as receiving incompatible data types. This approach involves the use of the same technologies already defined and evaluated in Section 4.1.

5. Development of web-application

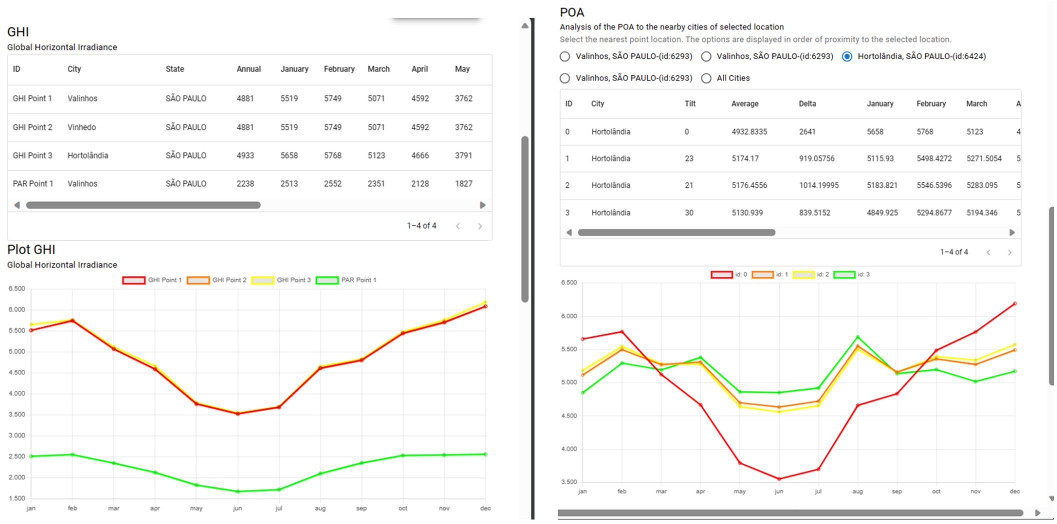
In this section, the implementation carried out for the project will be presented, which is the development of a Single Page Application (SPA) that allows the visualization of irradiance data using openly available databases, as well as prioritization of information extracted and fostered by national surveys.

Within each component of the system, Back-end and Front-end, they are gradually building a robust platform. The Front-end represents the client-side layer, while the other components form the server-side layers. To alleviate concerns about establishing a scalable system capable of supporting both horizontal and vertical expansion [Susan J. Fowler 2017], the decision was made to implement a microservices architecture in this work. Microservices are independent, smaller applications that are easier to extend features and improve testing quality.

The software development process began with the analysis of existing platforms and proceeded with the compilation of essential system requirements, considering the basic perspective of analysis of the PV plant and the approaches of three examples of web applications were presented providing relevant data in the photovoltaic sector, particularly focusing on irradiance data sharing. For the first feature to be implemented, could be called "Basic modelling and data visualization", because the focus of this stage is to create the basis of architecture at the same time provide a simple functionality as the initial tool. In this case, it applies the actual database available: Monthly mean GHI; Monthly mean POA; and Irradiance angles in 2-dimensional abacuses.

The navigability starts with the selection of the coordinate that can be a selection on the map or a numeric input field; following by the choice of available and connected databases that could be retrieved to the chosen region. The website collects the coordinates provided by the user and transfers them through API connections. Within the back-end layer, there are direct connections between the Database and the Django code, enabling the selection of appropriate data for each module and its corresponding API request. The back-end sends the response to the front-end, which adjusts its data according to the layout to be presented. Thus, the page undergoes a modularized update where only the items connected to the event change their current state.

Figure 3(a) showcases the layout section dedicated to presenting GHI data. The data is displayed in two intuitive formats: tables and graphs. The graphs feature distinct curves for each element and are interactive. In this stage, no mathematical modelling choice was involved in GHI data, only the sum of irradiance for the specific month and year is presented. If connected with the diagram presented in flowchart in Figure 1, these data can be provided by devices, such as pyrometers. Consequently, the monthly and annual data could be generated.



(a) Section of GHI. (b) Section of POA.

Figure 3. Layout web page feature 1.

Figure 3(b) represents the data, which accounts for irradiance considering the inclined plane. The data presentation format is the same as for GHI in Figure 3(a), with the added feature of filtering options to be selected for each region for visualization. This method of representation of the table added aims to simplify the comparison of tilt angles through the graph curves. In this stage, an arbitrary model, the model [Klein 1977] was adopted to validate the functionality of the web application focused on POA.

Figure 4 displays the third part of the web page, showing a contour histogram of the simulation of irradiance for the chosen coordinate, considering a list of variations in azimuthal angles and POA tilts [Santos et al. 2012]. This graph provides a more intuitive way to evaluate the most suitable angle range for a specific project involving the positioning of PV cells. The curve levels correspond to power output in kWh generated by the axis angles; In this figure, the range presented is 2 to 5 kWh.

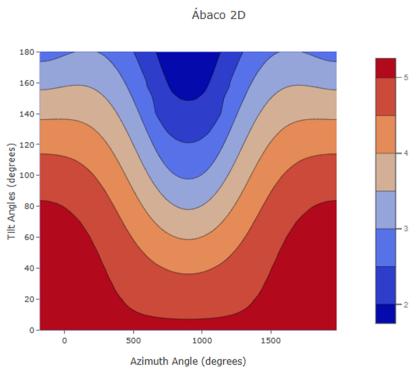


Figure 4. Layout web page, feature 1, Abacuses 2D.

The matrix is created employing rows and columns representing the Azimuth Angle and Tilt Angle of the photovoltaic module, respectively. The available intervals for the variation of the angles used in this analysis are from 0^o to 90^o for Azimuth, which uses its inclination with the reference north, while the tilt is between -180^o and 180^o which references the latitude. In this case, the same granularity was assumed for the two types of angles, 5 degrees of variations. In addition, for each matrix point, the model [Klein and Theilacker 1981] was applied to obtain the POA result.

6. Conclusion

Given the compatibility and usability issues in current software tools for PV energy, this work presents the development of an artifact purpose, as the web app to PV context. The scope of the system is based on the complete flow process of solarimetric data, enabling the simulation of real or fictitious photovoltaic plants. Considering this system, the first version of the application's structure was designed along with the evaluation of technologies, programming languages, frameworks, and other auxiliary tools of actual software designs. The resulting application allows a rich study of the PV plant allocation before and after its deployment, allowing its improvement. Another relevant result is that it is an application capable of being scalable as a microservice approach to the system evolution, considering the importance of many parallel sensors from PV plants. As future work, the purpose is to expand this web-based application with new features applied as microservices, considering distinct mathematical models to provide others with scientific analysis, such as benchmarking or suggestions to the optimized model to coordinate desired.

7. Acknowledgement

We would like to thank TotalEnergies for the financial support. In addition, we are grateful to all collaborators from Universidade Estadual de Campinas (UNICAMP). We acknowledge the support of ANP (Brazilian National Oil, Natural Gas and Biofuels Agency) through the R&D levy regulation. Acknowledgements are extended to the Center for Energy and Petroleum Studies (CEPETRO) and the School of Electrical and Computer Engineering (FEEC).

References

Bruno Montezano, Ana Paula C. Guimarães, and Marco A. Galdino (2017). SunData 3.0, CRESESB. http://www.cresesb.cepel.br/index.php?section=sundata&.

Bueno Pereira, E., Ramos Martins, F., Rodrigues Gonçalves, A., Santos Costa, R., J. Lopes de Lima, F., Rüther, R., Luna de Abreu, S., Máximo Tiepolo, G., Vitorino Pereira, S., and G. de Souza, J. (2017). *Atlas Brasileiro de Energia Solar*.

da Silva, M. K., Narvaez, D. I., de Melo, K. B., and Villalva, M. G. (2018). Comparative Analysis of Transposition Models Applied to Photovoltaic Systems Using Meteonorm and NASA SSE Databases. In *2018 13th IEEE International Conference on Industry Applications (INDUSCON)*, pages 237–241.

de Souza Silva, J. L., Barbosa de Melo, K., dos Santos, K. V., Yoiti Sakô, E., Kitayama da Silva, M., Soeiro Moreira, H., Bolognesi Archilli, G., Ito Cypriano, J. G., Campos, R. E., Pereira da Silva, L. C., and Gradella Villalva, M. (2022). Case study of photovoltaic power plants in a model of sustainable university in Brazil. *Renewable Energy*, 196:247–260.

de Souza Silva, J. L., Costa, T. S., de Melo, K. B., Sakô, E. Y., Moreira, H. S., and Villalva, M. G. (2020). A Comparative Performance of PV Power Simulation Software with an Installed PV Plant. In *2020 IEEE International Conference on Industrial Technology (ICIT)*, pages 531–535. ISSN: 2643-2978.

Fraczek, K. and Plechawska-Wojcik, M. (2017). Comparative Analysis of Relational and Non-relational Databases in the Context of Performance in Web Applications. In Kozielski, S., Mrozek, D., Kasprowski, P., Małysiak-Mrozek, B., and Kostrzewa, D., editors, *Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation*, Communications in Computer and Information Science, pages 153–164, Cham. Springer International Publishing.

Gaviria, J. F., Narváez, G., Guillen, C., Giraldo, L. F., and Bressan, M. (2022). Machine learning in photovoltaic systems: A review. *Renewable Energy*, 196:298–318.

Godboly, M. (2016). *Frontend Architecture for Desing Systems: A modern blueprint for scalable and sustainable websites*. O'Reilly Media, Inc., first edition edition.

Gray, J. L. (2010). The Physics of the Solar Cell. In *Handbook of Photovoltaic Science and Engineering*, pages 82–129. John Wiley & Sons, Ltd. Section: 3 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470974704.ch3.

Hay, J. E. and McKAY, D. C. (1985). Estimating Solar Irradiance on Inclined Surfaces: A Review and Assessment of Methodologies. *International Journal of Solar Energy*, 3(4-5):203–240. Publisher: Taylor & Francis.

INPE and LABREN (2017). Atlas Brasileiro de Energia Solar. http://labren.ccst.inpe.br/.

Kalay, M. , Kılıç, B., and Sağlam, (2022). Systematic review of the data acquisition and monitoring systems of photovoltaic panels and arrays. *Solar Energy*, 244:47–64.

Klein, S. (1977). Calculation of monthly average insolation on tilted surfaces. *Solar Energy*, 19(4):325–329.

Klein, S. A. and Theilacker, J. C. (1981). An Algorithm for Calculating Monthly-Average Radiation on Inclined Surfaces. *Journal of Solar Energy Engineering*, 103(1):29–33.

Kozhirbayev, Z. and Sinnott, R. O. (2017). A performance comparison of container-based technologies for the Cloud. *Future Generation Computer Systems*, 68:175–182.

Lima, G. P. D., Albuquerque, B. F., Sakô, E. Y., Silva, J. L. d. S., Moreira, H. S., and Villalva, M. G. (2020). Avaliação de desempenho para usina fotovoltaica de 1 MWp em Santa Rita do Sapucaí - MG. *Congresso Brasileiro de Energia Solar - CBENS*.

Plechawska-Wójcik, M. and Rykowski, D. (2016). Comparison of Relational, Document and Graph Databases in the Context of the Web Application Development. In Grzech, A., Borzemski, L., Światek, J., and Wilimowska, Z., editors, *Information Systems Architecture and Technology: Proceedings of 36th International Conference on Information Systems Architecture and Technology – ISAT 2015 – Part II*, Advances in Intelligent Systems and Computing, pages 3–13, Cham. Springer International Publishing.

Python (2024). WebFrameworks - Python Wiki. https://wiki.python.org/moin/WebFrameworks.

Santos, , Rüther, R., Nascimento, L., and Junior, L. (2012). Ábacos para análise simplificada de orientação e inclinação de sistemas solares fotovoltaicos integrados a edificações. *IV Congresso Brasileiro de Energia Solar e V Conferencia Latino-Americana da ISES*, pages 1–9.

Solargis, World Bank, and Energy Sector Management Assistance Program (ESMAP) (2019a). Global Solar Atlas. https://globalsolaratlas.info/.

Solargis, World Bank, and Energy Sector Management Assistance Program (ESMAP) (2019b). Global Solar Atlas 2.0 : Technical Report. Technical report. https://documents.worldbank.org/en/publication/documents-reports/documentdetail/529431592893043403/Global-Solar-Atlas-2-0-Technical-Report.

Soltesz, S., Pötzl, H., Fiuczynski, M. E., Bavier, A., and Peterson, L. (2007). Container-based operating system virtualization: a scalable, high-performance alternative to hypervisors. *ACM SIGOPS Operating Systems Review*, 41(3):275–287.

Susan J. Fowler (2017). *Microsserviços prontos para produção: Construindo sistemas padronizados em uma organização de engenharia de software*. novatec.

UNFCCC (1997). Kyoto Protocol Reference Manual on Accounting of Emissions and Assigned Amounts | UNFCCC.

Villalva, M. G. (2015). *Energia solar fotovoltaica: conceitos e aplicações*. Érica, São Paulo, 2 edition.

Whitaker, C. M., Townsend, T. U., Razon, A., Hudson, R. M., and Vallvé, X. (2010). PV Systems. In *Handbook of Photovoltaic Science and Engineering*, pages 841–895. John Wiley & Sons, Ltd. Section: 19 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470974704.ch19.

Zammetti, F. (2022). *Modern Full-Stack Development: Using TypeScript, React, Node.js, Webpack, Python, Django, and Docker*. Apress, Berkeley, CA.