Solarization through Google Engine Solar API **Handbook**

Google

icm
INICIATIVA CLIMÁTICA DE MÉXICO

Solarization through Google Engine Solar API **Handbook**

Mexico MMXXI

Authors

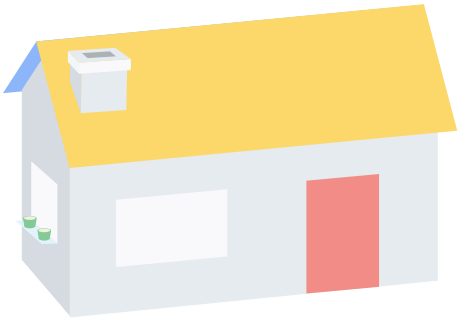
Isaura Espinosa de los Monteros Hinojosa

Ricardo Rubén Cruz Salinas

Angélica Valdiviezo Issa

Editorial Design:

Angel Armando Moreno Benitez



Google

icm
INICIATIVA CLIMÁTICA DE MÉXICO

Foreword

Iniciativa Climática de México (ICM) is a is a re-granting, think tanking, advocating, and project-implementing Mexican non-profit tax-free organization. ICM provides support to decision-makers and other key stakeholders (private sector, government, academy/research) to develop, implement and advocate for climate change and GHG emission-reduction public policies. ICM works to catalyze international climate policy at the national and city levels to reduce emissions of greenhouse gases in Mexico.

The handbook, which was developed by ICM, aims to be relevant for public policy makers, energy consultants, solar project developers or anyone interested in designing Massive Solar Distributed Generation Projects (MSDGP) using the Google Earth Engine Solar API. The document shows the experience of ICM elaborating a tool to identify potential solar users in Mexico and translates how the experience can be used for MSDGP. It has the aim of providing guidelines for the design of projects with Google Earth Engine Solar API as the main tool for evaluating solar characteristics. However, the handbook does not intend to be prescriptive on how to use the Google Earth Engine Solar API, hence, it describes the design of MSDGP as a venture that includes different aspects and proposes a guide to engage in each aspect.

The handbook is also limited to the design of the project and does not explore key areas of MSDGP as commercial and legal aspects or project implementation, including types of grid connecting and operations and maintenance (O&M).

The handbook is divided in four parts.
Part 1 – Describes the Google Earth Solar Engine API, its usage, and the data it provides.

Part 2 – Provides and overview of the existing software packages that can be used in designing a PV Project and how does the tools that can be created via the Google Earth Engine API present and advantage in designing a MSDGP.

Part 3 – Focuses on ICM experience using the API to create a web tool for the selection of beneficiaries of a Solar Residential Program. It is not the intention of the handbook to describe at great lengths the solar program for which the tool is planned, but only to describe the relevant elements that are needed to filter the solarization data accordingly.

Part 4 – Identification of best practices and recommendations to follow before and during designing a MSDGP with the API.

Acknowledgements

The publication titled *Solarization through Google Engine Solar API – Handbook* was produced by ICM, as part of its work on Photovoltaic Distributed Generation (PVDG) Technology and the development of a Program to increase the number of beneficiaries of PVDG Technology in Mexico from low- and medium-income households. The project was supported by Google.org and is one of the outcomes of a larger collaboration to retain additional capacity to develop solarization projects, measure and reduce greenhouse gas (CO_2) emissions and address energy poverty through a positive impact on household welfare.

Content

Figures

List of Abbreviations

API Application Program Interface

ICM Iniciativa Climática de México

kWh Killowatt hour

MSDGP Massive Solar Distributed Generation Projects

NE Northeast

PV Photovoltaic

PVDG Photovoltaic Distributed Generation

SW Southwest

Part **1**

GOOGLE EARTH ENGINE SOLAR API

What is an API?

An *Application Programming Interface* or API is a set of protocols that define how a user and an application or website communicate with each other. The purpose of this communication is to send information back and forth between the user and the application or website. The API establishes rules and procedures to *request* that information and to ensure the user can understand the *response*. In other words, the API lets the user know what is the correct way to *request* information to the application or website and how the *response* will look like. Moreover, an API can have one or more endpoints —or ends of a communication channel between the API and the user. Each endpoint can offer one or more resources that will allow for the transfer of data between the API and the user. For example, a governmental API could provide information about bus routes in a city. One endpoint could provide information about the bus routes and another could allow authorized users to save new information about the buses.

What is vector data? What is raster data?

Vector and raster are two types of spatial data. On one hand, vector data represents data of a region using geographical figures such as dots, lines o or polygons. For example, a dot might represent the location of a bus station, a line might represent the route of a bus, and a polygon might represent the city where the bus operates. Some data of these features —the bus, the route and the city— is represented by the geographical figures —a dot, a line, and a polygon. On the other hand, raster data represents spatial data through a matrix of squared cells (or pixels). Each of these cells or pixels contains information, such as the location, of the features. For example, a raster image could be a matrix of cells where each pixel indicates whether that area is part of a bus station or not; this raster image would be known as a bus station mask which spatially shows the location of the bus station within a geographical region.

The Google Earth Engine Solar API is a Google tool that provides users with rich solar potential data about buildings throughout the world. This data allows the users to understand in depth the characteristics of the buildings' roofs, the energy production for a wide range of proposed solar installations, and some financial information about the installation. All this information can prove valuable for users working towards the design and implementation of MSDGP.

To obtain data from the API, the user can follow the general process described by Figure 1. First, the user makes a request for information. As part of the request, the user needs to provide *request arguments* or parameters that specify the characteristics of the request based on the users' needs. The second step consists in the response from the API. If the request arguments are valid and if the API has information for the specific request made, then a valid response will be returned; otherwise, the response will be invalid.

The Google Earth Engine Solar API has two endpoints: the Vector Data Endpoint (*buildings*) and the Raster Data Endpoint (*solarInfo*). The *findClosest* endpoint provides technical and solar information about the building closest to a location of interest whereas the *get* method from the raster endpoint provides technical and solar information about a region surrounding a location of interest. The following sections cover each endpoint more in depth.

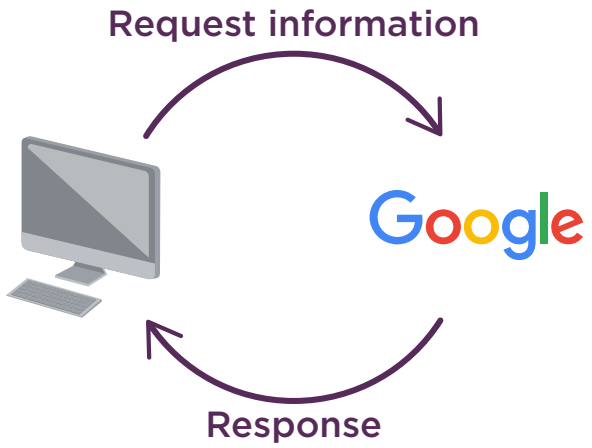


Figure 1. Simplified interaction between a user (computer) and an API (Google API)

Vector Data Endpoint (*buildings*)

The Vector Data Endpoint, *buildings*, provides solar data about buildings. More specifically, this endpoint has only one method: *findClosest*. This method provides solar data about the building closest to the pair of coordinates provided as part of the request. In other words, the *findClosest* method takes a location and returns the coordinates, dimensions, and solar potential information of the closest building to the given location; if there are no buildings within approximately 50 meters of the query point, the method returns an error that indicates that no buildings were found.

The request arguments for *building:findClosest* include the location, provided by a latitude and longitude. The method finds the closest building to the coordinate provided. As previously mentioned, an error will indicate if no buildings are near the provided location. On the other hand, a valid response will provide detailed statistics about the building and the rooftop, as well as solar data. The solar data provides a detailed description of the solar potential of the building through a set of possible configuration options for solar panels to be installed, as well as technical characteristics for each roof segment and each proposed panel. These configuration options vary in the number of segments and the number of panels per segment. For example, a possible configuration could consist of 4 panels distributed in 2 segments: one segment of 3 panels that would generate 1540.7 kWh yearly and another segment of 1 panel that would generate 509.1 kWh yearly. A second possible configuration for the same building could consist of 5 panels distributed in 2 segments: the same segment of 3 panels that would generate 1540.7 kWh yearly and another segment of 2 panels that would generate 1018.1 kWh yearly.

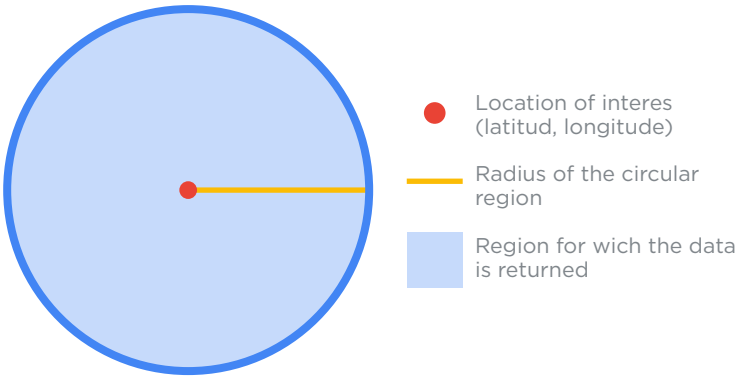


Figure 2. Sample region for which the API returns data

Raster Data Endpoint (*solarInfo*)

The raster endpoint or *solarInfo* resource provides information about the solar potential of a region surrounding a chosen location. More specifically, the *get* method takes a location of interest and a radius to define the region surrounding the former location; the method will return data for the region defined with these two parameters, as illustrated by Figure 2. The returned data consists of images that contain raster data for the region.

The raster data returned can be divided in two themes: descriptive and solar data. First, the descriptive data provides information of the region, such as the Digital Surface Map, an aerial photo, and a building mask to indicate which pixels of the image are part of the building. Second, the solar data provides information about the flux map and the hourly shade data. The flux map provides information about the amount of sunlight on the roof of the building and presents these data annually and monthly. The hourly shade of the region allows the user to establish whether a given cell in a region saw sun at a defined hour of a specific day of the year.

Interpreting data

This section presents the physical interpretation of the data returned by the API. It is worth noting the data usefulness depends on the characteristics of each MSDGP; for that reason, not all the data was used in the Hogares Solares project. As the data is continually updated, please refer to the Google Engien Solar API documentation for the most updated information.

As the previous section said, the API offers two endpoints; the *buildings* endpoint returns information about the building closest to a given location and the *solarInfo* endpoint provides information about the region surrounding a chosen location. Both endpoints return data in nested structures, which provides a logic to the data.

What is nested data?

Nested data refers to the structure that organizes several pieces of data. To provide more clarity, nested data has several layers of information and each layer can be made of objects that contain objects themselves. The representation of these data structures varies across programming language, but the essence is generic. For example, a user can come across data that describes a bus route. The first layer can contain basic statistics such as the daily average number of passengers and the total length of the route, as well as nested objects such as a list of buses that transit that route and an object that describes in detail the buses. Figure 3 illustrates a representation of an example of this nested data structure.

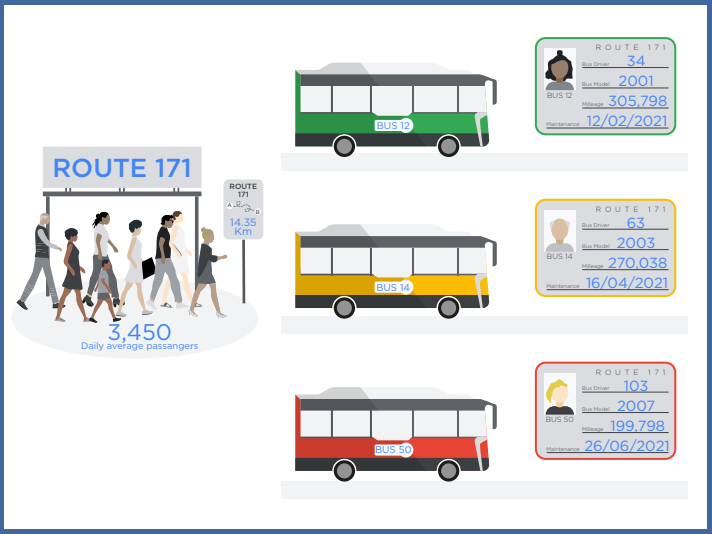


Figure 3. Sample nested data

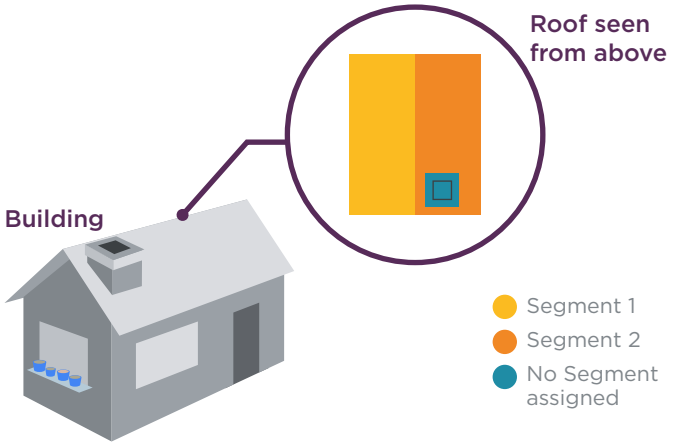


Figure 4. roofSegmentStats, information to describe each roof segment.

Buildings

The first kind of information contained in every response of this type of request is about the location of the closest building. First, the name of the building, assigned by the API. Then, there are the coordinates for the center of the building, and for its most southwest (sw) and northeast (ne) points, contained in the boundingBox of the building. For the US, more information about the location is given by the postalCode, the administrativeArea, the statisticalArea, and the regionCode.

As Google Earth Engine Solar determines the solar potential by processing imagery data, the date when such imagery was obtained is given by the imageryDate. The results of such processing are contained in the solarPotential object and for easier understanding, here are classified in three sets: the roof, the photovoltaics, and the financial.

THE ROOF

Google Earth Engine divides the roof in different segments, then each segment is analysed to obtain the solar potential. Hence, the roof is characterized by its segments and their physical characteristics are presented in the roofSegmentStats (see Figure 4) which is an

Programming language is marked in blue.
Ej. imageryDate

object that contains the information to describe each roof segment presented as an array. In other words, the first roofsegment is in the position 0 of the array, the second one is in the position number 1, etc. The object may be looked as the following:

roofSegmentStats : (segment 1, segment 2, segment 3, ..., segment N)

Position in the arrey	0	1	2	N–1

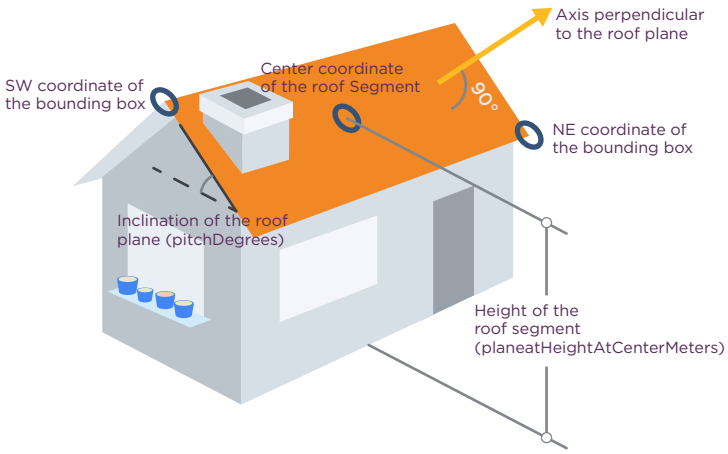


Figure 5. planeHeightAtCenterMeters, necessary data to characterize the roof

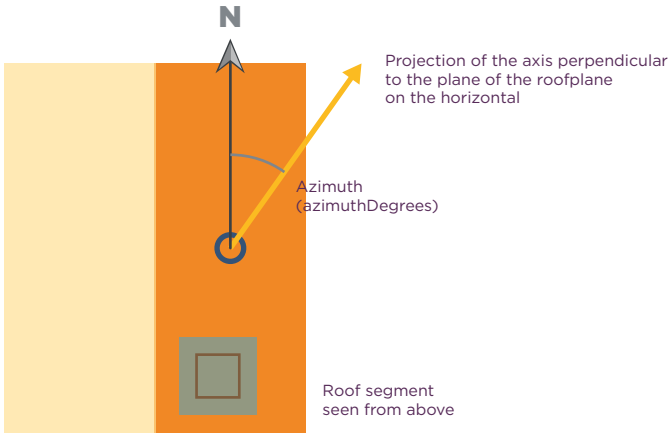


Figure 6. azimuthDegrees, the azimuth axis

Each position of the array contains all the necessary data to characterize the roof segment: its height in meters above the ground (planeHeightAtCenterMeters) (see Figure 5), its inclination against the horizontal plane (pitchDegrees), and the cardinal direction where the perpendicular axis of the roof plane (the azimuth axis) is facing —North=0°, East=90°, and so on— (azimuthDegrees) (see Figure 6). For each segment, its center and boundingBox coordinates are also given, as well as the stats containing the area of the roof segment areaMeters2, its horizontal area projection (groundAreaMeters2) (see Figure 7), and the sunshine quantiles (sunshineQuantiles). These quantiles are the result of measuring the sunniness of N points across the roof segment. For N measures there will be N-1 quantiles.

Due to shades or other features of the imagery, not every part of the building may be assigned to a roof segment. For those assigned to a segment, a summary is provided in the wholeRoofStats which contain the areaMeters2, that is the sum of all the roof areas, accounting for tilt, so it may be higher than the ground footprint area of the building (which is in the groundAreaMeters2) and the sunshineQuantiles. The buildingStats presents the same information as the wholeRoofStats but including those unassigned parts.

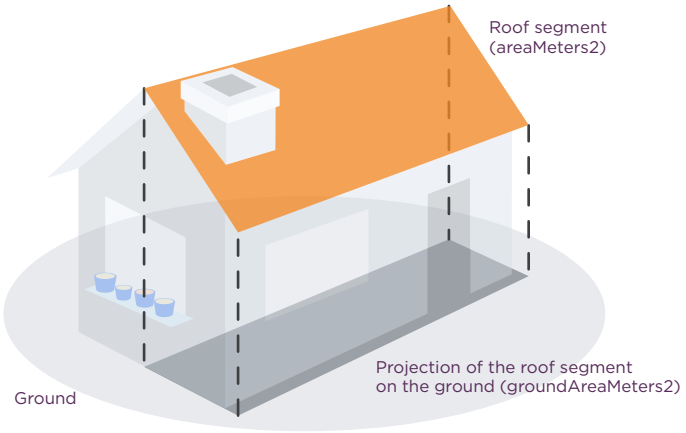


Figure 7. groundAreaMeters2, area of the roof segment, horizontal area projection

THE PHOTOVOLTAICS

On this matter, both the solar panels and their energy output are presented. For the solar panels it is important to know by, up to date, Google Earth Engine Solar only considers one single type of device that has a nominal capacity of 250 W. Its dimensions are 2 meters height, 1 meter width. These characteristics, including its lifetime are presented in the panelCapacityWatts, panelHeightMeters, panelWidthMeters, and panelLifetimeYears.

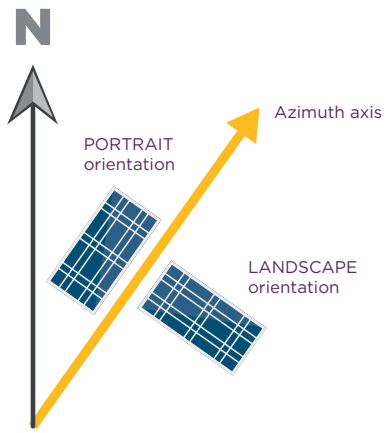


Figure 8. Panel orientation

Each solar panel location is presented in the solarPanels. It includes the center coordinates for its installation (center) and the indication if its long edge should be perpendicular or parallel to the azimuth (orientation) (see Figure 8), the segment where it would lie (segmentIndex), and the energy that it would generate over the course of a year (yearlyEnergyDcKwh).

The API shows the information about different possible ways to install the solar panels —known as configurations— in a particular way that is compiled within the solarPanelConfigs that is an object that presents the information of one configuration in one position of the array. Each configuration is described by the total amount of solar panels to install (panelsCount), the energy generated by the whole configuration (yearlyEnergyDcKwh), and an object called roofSegmentSummaries that relates the panels of the configuration to the segments needed for their installation. For doing so, the roofSegmentSummaries is an array with the information of one segment in one of its positions. It presents the inclination of the roof segment (pitchDegrees), its azimuth (azimuthDegrees), the number of panels that it contains (panelsCount), and the position where this roof segment lies inside the roofSegmentStats object (segmentIndex). Ultimately, it is with

segmentIndex that the position of the panels in the configuration can be known. The first position of the roofSegmentSummaries array contains the information of that configuration with the least number of panels. The second position has the information of the configuration that follows the first one in the amount of panels. Hence, the last position in the array corresponds to the configuration that has the maximum number of panels that may be installed on the building. This number is also presented in the maxArrayPanelsCount and the area such configuration would occupy is in the maxArrayAreaMeters2.

solarInfo

The *solarInfo* endpoint returns conceptually different data; the main difference is that the data obtained are mainly URLs to fetch the images (in GeoTIFF formats) that contain the raster data. It is worth noting that the URLs are valid for a limit of time after they are generated by the request, which is why it is recommended to access the data in the GeoTIFF files as soon as possible after requesting them. Moreover, the user can determine whether to obtain all or certain images from the request, as explained in the documentation.

As previously mentioned, the solar potential is determined by processing imagery data. Just as the *buildings* endpoint, the *solarInfo* enpoint also provides the date when such imagery was obtained (imageryDate) as well as the date when the imagery processing was completed (imageryProcessedDate).

As a clarification, all the GeoTIFF files are at a resolution of 10cm/pixel, except for the monthly flux file (with a resolution of 50cm/pixel) and the hourly shade files (with a resolution of 1m/pixel).

DESCRIPTIVE RASTER DATA

The first descriptive image is the Digital Surface Map (DSM) of the region formed by the request; the URL to access its GeoTIFF file is provided in the field dsmUrl. The user must then access and process the raster data. The DSM provides information for the meter above the sea level for the region. The second descriptive image is the aerial

photo provided by the URL in the field rgbUrl. This image contains an RGB photo of the region. The third descriptive image is the building mask, provided by the URL in the field maskUrl. This mask let the user know whether each pixel is part of a rooftop or not.

SOLAR RASTER DATA

The solar raster data is provided by three fields. The first field, annualFluxUrl, provides a URL from which the user can fetch the annual flux map. This map provides information about the annual sunlight on roofs measured in kWh/kW/year. This map allows the user to know the amount of sunlight a pixel (10 x 10 cm) received in the year.

The second field, monthlyFluxUrl, provides a URL for a GeoTIFF file at a resolution of 10 cm/pixel. Similar to the annual flux map, this map provides monthly flux data for the region measured in kWh/kW/year. Unlike the annual flux file, this GeoTIFF contains 12 bands, one for each month of the year. For example, using the second band of the GeoTIFF, the user can obtain the amount of sunlight for each pixel (50 x 50 cm) of the region for February.

The final field, hourlyShadeUrls, provides even more granular solar data for the given region at a resolution of 1m/pixel. This field provides a list of 12 URLs, one for each month of the year. Each URL (representing a month) allows the user to fetch a GeoTIFF file which, in turn, contains 24 bands (or channels) corresponding to the 24 hours of a day. Finally, each pixel contains data to know whether a given location within the region was able to see the sunlight that day, hour and month of the year. The documentation provides an example to understand how to access specific information from these URLs:

If you want to know whether a point (at pixel location (x, y)) saw sun at 4pm on the 22nd of June you would:

1. *fetch the sixth URL in this list (corresponding to June).*
2. *look up the 17th channel (corresponding to 4pm).*
3. *read the 32-bit value at (x, y).*
4. *read bit 21 of the value (corresponding to the 22nd of the month).*
5. *if that bit is a 1, then that spot saw the sun at 4pm 22 June.*

Part **2**

SOLAR RESOURCE TOOLS

Currently, there are many different software packages dedicated to estimate solar potential and to evaluate the feasibility of a project. As each project has its own objectives, more than one tool may be required for an effective design. The objective of this section is to present the general characteristics of other solar tools available for a better understanding on how to make use of Google Earth Engine Solar API.

There are three main aspects to focus on when designing a PV project:

1. the solar radiation and the meteorological (solar data);
2. the PV performance, relevant to the system, and;
3. the financial aspects.

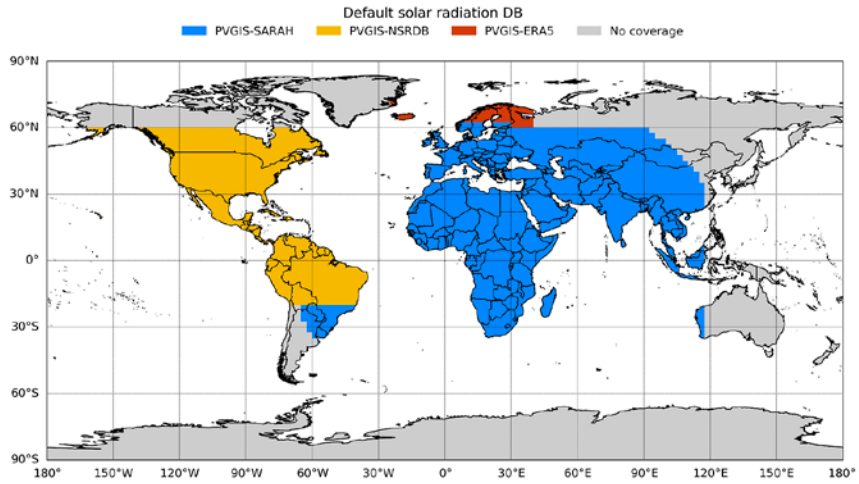


Figure 9. Solar radiation databases available across the world. Taken from the PVGIS users manual, available at https://ec.europa.eu/jrc/en/PVGIS/docs/usermanual

Solar data

All software packages that assist on the design of PV projects, rely on solar data to estimate the energy production of the project. These data usually come from satellite or local measurements of metrics like solar irradiance, temperature, air pressure, wind speed, among others. Usually, the measurements are done using satellites and the data is compiled in many datasets like NASA's MERRA[1]. Some entities have compiled several datasets and offer regional data sets like the National Solar Radiation Database (NSRDB) by the National Renewable Energy Laboratory (NREL)[2]. Figure 9 shows the default databases from the web application called PVGIS[3].

1 https://gmao.gsfc.nasa.gov/reanalysis/MERRA/
2 https://nsrdb.nrel.gov/about/what-is-the-nsrdb.html
3 https://ec.europa.eu/jrc/en/PVGIS/docs/usermanual

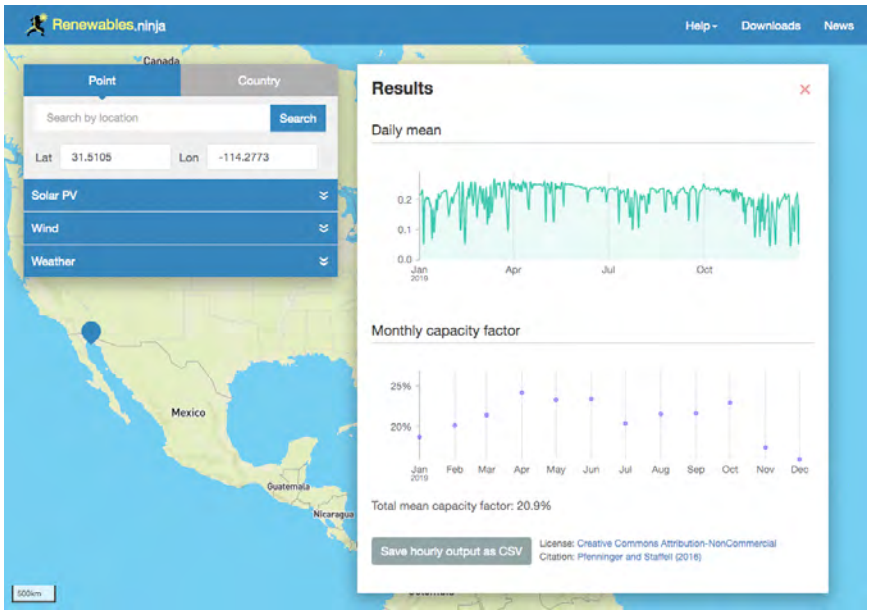


Figure 10. Screenshot of the Renewables.ninja solar data tool. Available at https://www.renewables.ninja/

Almost every software uses an interactive map to select the location (using coordinates) for data acquisition. Some even have implemented API based data access. Figure 10 shows ninja.renewables[4] tool that gives solar, wind or meteorological data for a given year. As it can be seen, the data is at hourly level.

PV performance

PV software packages use solar data to calculate the energy output from a system that the user may characterize depending on the objectives of the project. Many platforms allow the user to select the type of solar panel, capacity, system losses, and mounting characteristics. Based on these, the software estimates the yearly, monthly, or hourly electric generation of the system. Almost every software has the option to select the location using an interactive map, coordinates or even an address. Figure 11 shows the PVGIS[5] interface. One can choose between grid connected, tracking PV, or off-grid systems.

Some tools have the capability to draw the area where the system could be placed and so, to estimate the capacity of the system, instead of having it as an exogenous data (see for example Figure 12 in which PVWatts[6] allows the user to draw the available area on an interactive map).

The most sophisticated tools not only allow the user to draw the area, but also to place each solar panel, wires, inverters, and other equipment with a Computer Assisted Design (CAD) tool, so the engineering design of the entire system can be done. Is this kind of software that the utility scale developers use. Figure 13 shows some examples of diagrams for system design using Helioscope[7].

4 https://www.renewables.ninja/
5 https://re.jrc.ec.europa.eu/pvg_tools/en/#PVP
6 https://pvwatts.nrel.gov/pvwatts.php
7 https://www.helioscope.com/

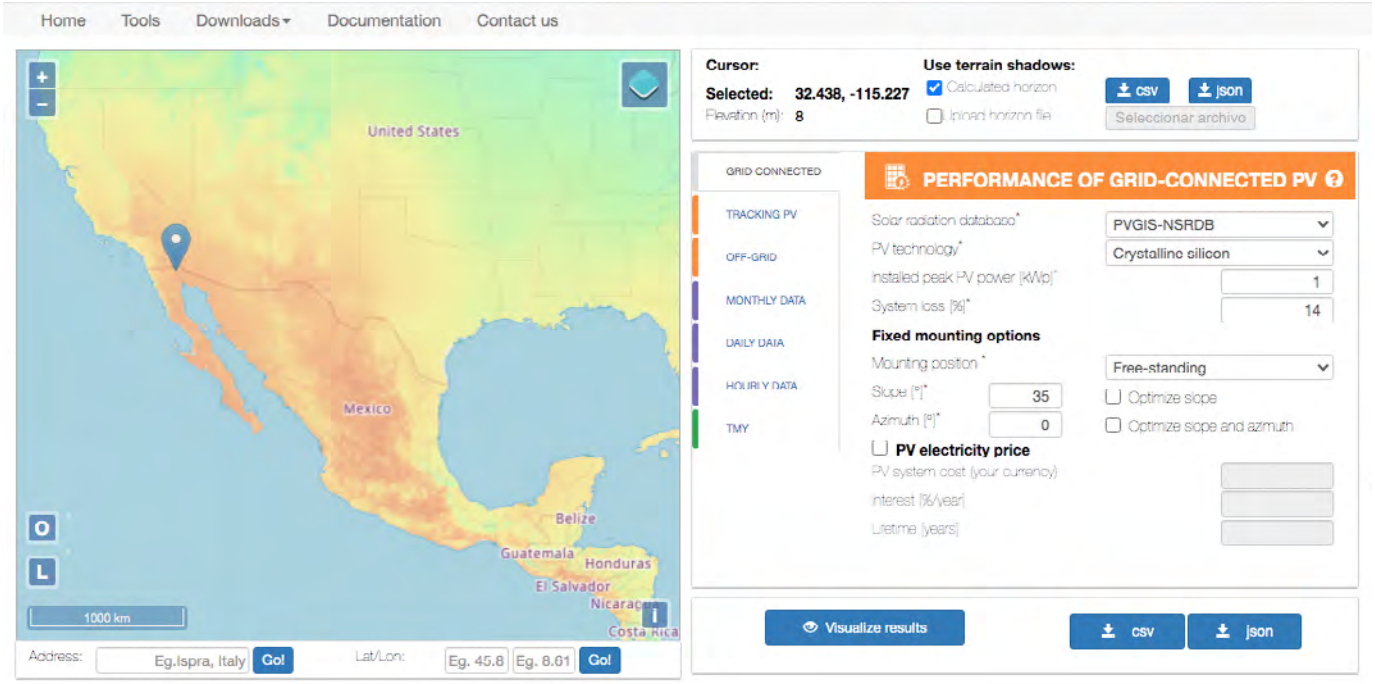


Figure 11. Screenshot of the Photovoltaic Geographical Information System, available at https://re.jrc.ec.europa.eu/pvg_tools/en/#PVP

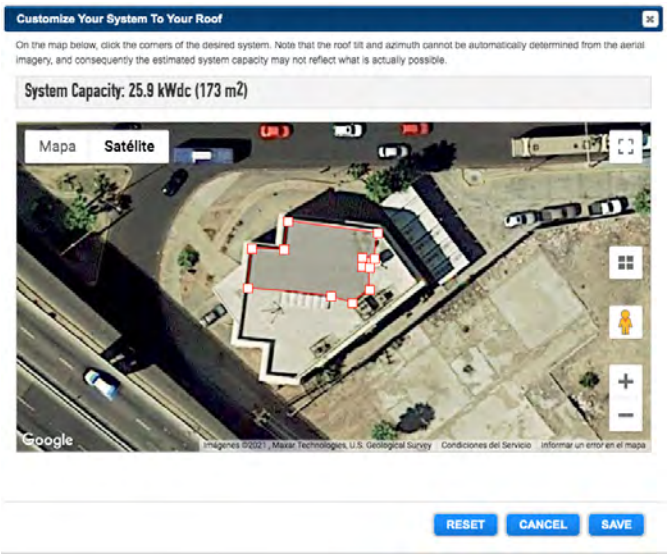


Figure 12. Screenshot of the "Draw your system" feature of the PVWatts tool. Taken from https://pvwatts.nrel.gov/pvwatts.php.

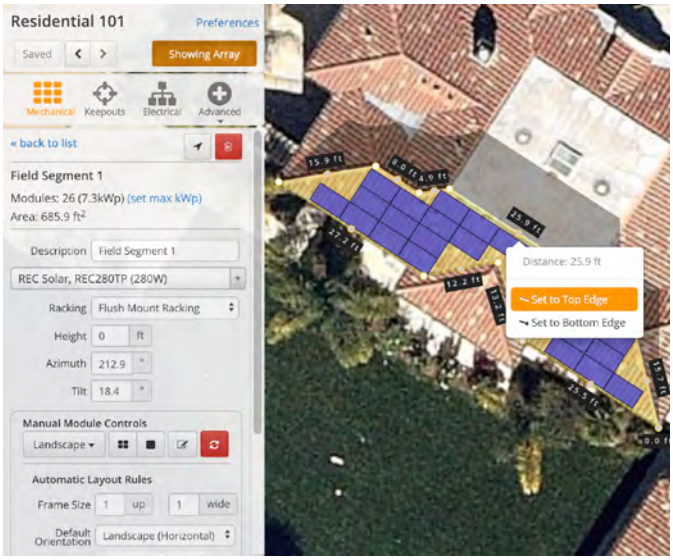


Figure 13. Example of the solar panels placement for the design of a residential project using Helioscope. Taken from the Step by Step Residential Design guide of Helioscope, available at https://help.helioscope.com/article/36-residential-overview.

Financial

Different tools have incorporated some form of financial analysis to evaluate the feasibility of the project. The financial tools incorporate the system cost at the moment in time and place, its lifetime, and an interest rate to calculate the system cost per each generated kWh. Some tools include a feature to take the grid electricity costs to estimate the value of the energy output. And there are other software packages that offer financial figures like the internal rate of return, the return of the investment, or the net present value.

The features of Google Earth Engine Solar API for MSDGP

Usually, the process described in the previous section is followed by a photovoltaics developer to design a project for each of its clients. No matter if designing a utility scale power plant with hundreds of panels, or just a residential distributed generation system with nearly ten, the developer focuses just on one system at a time. But what happens if a developer, government, or any other stakeholder desires to design a project for more than one facility at a time? Moreover, what if the project focuses on analyzing the feasibility of solar installations of hundreds or thousands of solar systems? Clearly, this process would take a long, long time.

Google Earth Engine Solar API offers a convenient solution to design a massive project, as the one can program a tool to send requests to the API and then to examine the response in order to filter those buildings that have the adequate characteristics for the MSDGP. The following chapter will describe how ICM programmed a web tool for the Hogares Solares project.

Through the API the tool can automatically estimate the usable area of the roof of any building. Its algorithm, by segmenting the roof, identifies which areas may accommodate a solar panel and how it could be placed. Additionally, it identifies the orientation and inclination of that segment and how much solar irradiance it receives, so to calculate the energy output of the panel. Because of all of these traits the Google API has and all the information it reports back

to the user, using other sophisticated designing software to draw an estimate the capacity by simulating placing the panels becomes unnecessary.

Finally, since Google Earth Engine Solar API algorithm processes the imagery of the building, it is capable of estimating the solar potential of the very point based on its physical characteristics. As a result, when many buildings within an area are evaluated, one can be sure that its solar potential is specific for each one and is not taking one same aggregated estimation like in the solar data from a satellite.

The following table is a comparison between the Google Earth Engine Solar API and the other software packages compared in this chapter.

	Google Earth Engine Solar API	Solar tools	Professional solar tools
Metheorological data (temperature, air pressure, wind speed, etc)	No	No	Selection of databases (background data)
Solar irradiance	At a pixel level (1 square meter). May be processed by the user	Several meters wide. Background data.	Several meters wide. Background data that may be selected by the user.
Calculation of PV performance	Yes	Depending on customizable basic options	Full customization of the installation and materials
Physical project definition	Building carachteristics given by the software	No	Users may draw the system
Number of installations per project	Massive	1	1
Main use of the software	Photovoltaic potential in the cities, pre-design of a massive number of photovoltaic projects	Pre-design of a photovoltaic project	Design of a photovoltaic project

Part **3**

DESIGNING A MSDGP. SELECTION CRITERIA

Hogares Solares is a project with the objective to reduce both energy costs and climate impact of the residential electricity users by providing them with solar panels connected to the grid (distributed generation). For the project to be financially sustainable, there are fundamental criteria that must be met in the selection of the benefited residential electricity users.

Criteria for Hogares Solares project

The project aims to give photovoltaic systems to low- and middle-income residential electricity users. By themselves, the PVDG Systems are not affordable for these households. Thus, the Hogares Solares Project, creates a mechanism to loan the systems by re-canalizing subsidy and reducing the beneficiaries' electricity bill thanks to the energy produced by the solar panels.

For the program to be successful and provide benefits, for the users, the utilities and the environment, the users and their dwellings had to comply with certain characteristics. The following is a list of characteristics that the users and their dwellings must have:

- User must be the owner of the house and the house must hold only one household.
- No debts to the utility.
- The dwelling must have less than two stories and the roof must be of concrete to maintain the system costs in the range of profitability.
- Consumption must be around 2,500 kWh and 3,500 kWh per year. This sets the user in the range of the profitability of the system costs.
- The house building has to have the solar potential to supply the user energy consumption. In other words, the house must have enough space to adjust a given number of panels to produce the required power. In the case of this project, and including the costs estimates, the number of solar panels was set in 5 to 6.

Criteria evaluation with Google Earth Engine Solar API

As the project is intended to be implemented in some cities of Mexico, Google Earth Engine Solar API was used to filter all the buildings with enough solar potential to satisfy the energy consumptions of 2,500 to 3,500 kWh per year.

A program was developed to send requests to the API within a geographical area defined by the user. For Hogares Solares, sociodemographic data was key, which is why the geographical data used were administrative blocks as defined by INEGI, the Mexican National Institute of Statistics and Geography (INEGI). The program created a grid within the blocks within the chosen city and sent requests from each point to ensure that the whole area was covered with the requests. For each request, the API sends the information of the closest building -as described in the sections above.

Then, the program processed the responses to select the "best" segment given the program's needs. To make this choice, the program analyzes each of the segments of each of the solar panel configurations in solarPanelConfigs. To simplify installation, the program chooses the "best" segment of all the segments of all the proposed solar configurations. The "best" segment is selected as follows. Aligned to Hogares Solares selection criteria, the eligible segments are those that capture at least 3,532 DC kWh yearly as per the yearlyEnergyDcKwh field (sunlight energy captured over a year) and that have a maximum of 11 panels (due to financial considerations). Zero or more segments can be eligible for a building. If no segments are eligible, then the building is discarded. When only 1 segment is eligible, that segment is considered the "best". When more than one segment is eligible, then the segment with the lowest pitchDegrees is selected also due to financial constraints.

If the program finds a "best" segment for the building, then the building is included in the list of eligible buildings along with the building's and segment's information. For the building, the dataset includes the following information: latitude, longitude and id. For the "best" segment, the following information: index, number of solar

panels, energy generation capacity installed, sunlight energy captured, azimuth, pitch degrees, height, latitude, longitude and bounding box. Moreover, each building also contains sociodemographic information about the administrative block where it is located. The next section presents such data more in depth.

Non-solar criteria

As the Hogares Solares Program has a sociodemographic component, the solar and financial data needed to be complemented. Thus, the geographical data from the National Housing Inventory provided information at the block level. The relevant information for Hogares solares was the proportion of inhabited housing units in the block, proportion of inhabited housing in the block that have electricity, street lighting surrounding the block, and paved streets surrounding the block. While none of this information was directly used to filter out buildings, the information provides the program implementers the ability to filter accordingly.

Part **4**

BEST PRACTICES AND RECOMMENDATIONS

The following are best practices and recommendations when using the Google Earth Engine Solar API to develop a MSDGP.

1 Clear definition of project objectives and criteria

A clearly defined objective will allow the project to efficiently access and use the data. Based on the documentation and the information presented in the manual, other users can define their goals, understand the data the API offers and then explore which data satisfies their needs. If the users invest enough resources and time to accomplish these steps, then they can achieve efficient access to the data. In contrast, an unclear definition of the objectives for the data can hinder progress and the users might face the need for do-overs.

2 Testing before requesting large amounts of data

The project will benefit from an adequate estimation of the resources needed to access the data from the API. Bigger geographical areas require more resources —especially time— to send the requests and to access, process and store the returned data. Regardless of the area size, all projects can benefit from testing; it can help estimate the resources needed to obtain the information for the whole area of interest, as well as to improve the design of the data wrangling and storing processes.

As an example, the project Hogares Solares tested its data acquisition process on a random sample of 5% of the territory it aimed to cover. This test allowed the team to: identify missing data that was indeed of interest and had not been saved; better quantify the computational need to store the total data to be accessed; and to build a reasonable timeline for the process.

3 Map of the data journey

Once the users understand better what data they need and why, a good practice is to map the journey of the data. Some of the most common steps are access, processing, storing and use by the end client. The Hogares Solares project mapped the data journey backwards, from the end use towards the access. This reversed approach provided feedback to the definition of objectives for the data as well as to identify loopholes where the data had not a clear path to reach the next step.

Additionally, mapping the data allowed us to make progress on the technical components of some of the end steps of the journey, such as the storage and the use by the final client. Every project has its unique logistical and administrative constraints and sorting them out on a timely manner can mitigate risks. Particularly, the project may need to store larger volumes of data which might benefit from using services on the cloud. Mapping those needs sooner can allow the users to plan efficiently for the future needs.

References

NASA. (2019). MERRA-2 - Global Modeling and Assimilation Office [Online]. Available at: https://gmao.gsfc.nasa.gov/reanalysis/MERRA/ (Accessed: January 29, 2021)

NREL. What is the NSRDB? [Online]. Available at: https://nsrdb.nrel.gov/about/what-is-the-nsrdb.html (Accessed: January 29, 2021)

EU Science Hub - European Commission (2020). PVGIS users manual [Online]. Available at: https://ec.europa.eu/jrc/en/PVGIS/docs/usermanual (Accessed: January 29,2021)

Renewables.ninja [Online]. Available at: https://www.renewables.ninja/ (Accessed: January 29, 2021)

EU Science Hub - European Commission (2020).PVgis [Online]. Available at: https://re.jrc.ec.europa.eu/pvg_tools/en/#PVP (Accessed: January 29,2021)

NREL. PVWatts Calculator [Online]. Available at: https://pvwatts.nrel.gov/pvwatts.php (Accessed: January 29, 2021)

HelioScope [Online]. Available at: https://www.helioscope.com/ (Accessed: January 29, 2021)

HelioScope. (2020). Step by Step Residential Design - HelioScope Knowledge Base [Online]. Available at: https://help.helioscope.com/article/36-residential-overview (Accessed: January 29, 2021)

Solarization through Google Engine Solar API **Handbook**

Mexico MMXXI

Google

icm
INICIATIVA CLIMÁTICA DE MÉXICO