



UNIVERSIDAD POLITÉCNICA DE MADRID  
ESCUELA TÉCNICA DE INGENIERÍA Y DISEÑO INDUSTRIAL  
Grado en Ingeneriería Eléctrica

---

TRABAJO DE FIN DE GRADO

**Aplicación web para la estimación de  
una instalación solar conectada a red**

Autor: Ionut Cristian Morariu

---

Tutor: Oscar Perpiñán Lamigueiro  
Departamento de Ingeniería Eléctrica,  
Electrónica, Automática y Física aplicada

Madrid, 19 de junio de 2020

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos . . . . .	1
1.2. Análisis previo de soluciones . . . . .	2
1.3. Aspectos técnicos . . . . .	3
1.3.1. Backend . . . . .	3
1.3.2. Frontend . . . . .	4
<b>2. Estado del arte</b>	<b>5</b>
2.1. Situación actual de la generación fotovoltaica . . . . .	5
2.2. Soluciones existentes y sus carencias . . . . .	6
<b>3. Parte teórica y desarrollo del código</b>	<b>10</b>
3.1. Naturaleza de la radiación solar . . . . .	11
3.1.1. Radiación fuera de la atmósfera terrestre . . . . .	12
3.1.2. Cálculo de componentes de radiación solar . . . . .	14
3.2. Radiación en superficies inclinadas . . . . .	16
3.2.1. Estimación de irradiancia a partir de irradiación diaria . . . . .	16
3.2.2. Transformación al plano del generador . . . . .	18
3.2.3. Pérdidas por ángulo de incidencia y suciedad . . . . .	21
3.3. Cálculo de la energía producida por el generador . . . . .	24
3.3.1. Definición de un SFCR . . . . .	24
3.3.2. Configuración de los elementos del sistema . . . . .	24
3.3.3. Funcionamiento de una célula solar . . . . .	25
3.3.4. Comportamiento térmico de un módulo . . . . .	26
3.3.5. Cálculo final de potencias y energías . . . . .	35
<b>4. Ejemplo práctico de aplicación</b>	<b>39</b>
4.1. Obtención de datos del usuario . . . . .	39
4.1.1. Latitud y longitud a partir de la dirección . . . . .	40
4.1.2. Datos de la superficie destinada a la instalación . . . . .	40
4.1.3. Obtención de la inclinación y orientación de la superficie . . . . .	41
4.2. Aplicación de los datos al proceso de cálculo . . . . .	46
4.2.1. Valores medios mensuales de radiación global . . . . .	47
4.2.2. Irradiancia extra-terrestre diaria . . . . .	47

4.2.3. Separación de la radiación global horizontal en sus componentes . . . . .	47
4.2.4. Irradiancia en la superficie inclinada . . . . .	51
4.2.5. Pérdidas por ángulo de incidencia y suciedad . . . . .	55
4.2.6. Configuración del módulo solar y su comportamiento . . . . .	58
4.3. Cálculo final de las potencias y energías . . . . .	65
<b>5. Detalles de la programación</b>	<b>68</b>
5.1. Descripción de tecnologías . . . . .	68
5.1.1. Servidor . . . . .	68
5.1.2. Cliente . . . . .	70
5.2. Explicaciones de código . . . . .	71
5.2.1. Emplazamiento del usuario . . . . .	71
5.2.2. Área, inclinación, orientación y nivel de suciedad de la superficie de instalación	73
5.2.3. Obtención de la información de radiación global en el plano horizontal . . .	73
<b>A. Código completo</b>	<b>79</b>
A.1. Código del servidor . . . . .	79
A.1.1. DataController.js . . . . .	79
A.1.2. Helpers.js . . . . .	87
A.1.3. SolarData.js . . . . .	94
A.1.4. StationsData.js . . . . .	95
A.1.5. Router.js . . . . .	96
A.1.6. App.js . . . . .	97
A.1.7. Start.js . . . . .	98
A.2. Código de cliente . . . . .	99
A.2.1. dataAquisition.js . . . . .	99
A.2.2. GenerateLinks.js . . . . .	111
A.2.3. index.html . . . . .	113
A.2.4. Results.html . . . . .	119
A.2.5. renderResults.js . . . . .	122
<b>Bibliografía</b>	<b>126</b>

# Capítulo 1

## Introducción

### 1.1. Objetivos

El objetivo de este proyecto es el desarrollo de una aplicación o página web de fácil acceso para todos los usuarios, con la finalidad de ofrecer una estimación inicial de la posible generación de una instalación fotovoltaica doméstica de conexión a red.

Durante el resto del documento, si fuera necesario, se hará referencia a la aplicación desarrollada en este proyecto con el nombre de SolarCalc.

Para poder llevar a cabo ésta estimación, el usuario introducirá unos serie de datos acerca de su emplazamiento y edificación en la que desea situar la instalación, y la aplicación hará todos los cálculos necesarios para ofrecer una aproximación lo más cercana al resultado final teniendo en cuenta todas la variables que puedan intervenir.

La aplicación también ofrecerá otros datos de posible interés para el usuario como: el número de paneles que se pueden instalar, la potencia de dichos paneles y la potencia del inversor.

La idea de esta aplicación surge de una conversación con un conocido cuando estaba planificando la construcción de su nueva vivienda, en la cual quería realizar una instalación fotovoltaica para reducir el gasto en la factura de electricidad.

En su búsqueda no encontró ningún servicio que fuera lo suficientemente sencillo de entender para una persona sin ningún tipo de conocimiento previo acerca de la generación fotovoltaica, que le aportase la posibilidad de poder personalizar los cálculos a su emplazamiento y planos de construcción.

Otro de los puntos clave de la aplicación es que sea de código abierto y gratuita para los usuarios, usando fuentes de información disponibles para cualquier interesado. Todos los pasos y operaciones se podrán obtener, analizar y reutilizar de manera gratuita desde un repositorio de Github<sup>1</sup>.

---

<sup>1</sup>Github.com: Plataforma online de almacenamiento de código y documentación de fuentes abiertas.

Los objetivos detallados de esta aplicación son los siguientes:

- Diseñar una interfaz de usuario amigable y sencilla de usar para que la pueda utilizar un gran número de personas sin necesidad de conocimientos sobre energía fotovoltaica.
- Obtención de los datos de radiación en el emplazamiento indicado por el usuario mediante el uso de API<sup>2</sup> externas.
- Realizar todos los cálculos necesarios para ofrecer una estimación competente de los siguientes datos:
  - Número de paneles que se pueden instalar.
  - Potencia máxima a instalar.
  - Potencia del inversor.
  - Energía eléctrica producida en un año.

## 1.2. Análisis previo de soluciones

Antes de comenzar el desarrollo del proyecto, se llevó a cabo una revisión de las soluciones de estimaciones de instalaciones fotovoltaicas existentes en el mercado para decidir si tenía cabida una aplicación como la que se iba a desarrollar.

Algunas de las soluciones encontradas fueron:

### 1. PVsyst - Photovoltaic Software

El software PVsyst, desarrollado por la empresa suiza con el mismo nombre es quizás el más conocido dentro del ámbito del estudio y la estimación de instalaciones fotovoltaicas. Ofrece una amplia capacidad de personalización de todos los componentes de la instalación.

### 2. CalculationSolar.com

Es el primer resultado de Google al buscar el término “calculadora de instalaciones fotovoltaicas”, por tanto será una de las primeras aplicaciones que una persona que desea realizar una instalación en su vivienda visite.

### 3. SISIFO

Es una herramienta web diseñada y desarrollada por el Grupo de Sistemas Fotovoltaicos del Instituto de Energía Solar de la Universidad Politécnica de Madrid. Ha sido y es la herramienta interna utilizada por los ingenieros de dicho grupo.

### 4. PVGIS

Aplicación web desarrollada por el European Commission Joint Research Center desde 2001. Su enfoque es asistir en el cálculo y estimación de instalaciones fotovoltaicas, ya sean conectadas a red, de seguimiento o de autoconsumo.

---

<sup>2</sup>Application Programming Interface: conjunto de funciones y procedimientos que ofrece la posibilidad de un software a interaccionar con otro.

## 5. System Advisor Model

System Advisor Model (SAM), desarrollado por el Laboratorio Nacional de Energías Renovables, perteneciente al Departamento de Energía del gobierno americano, es una software técnico-económico gratuito que ayuda a la toma de decisiones en el amplio campo de las energías renovables. Ofrece un conjunto de soluciones muy completas no solamente relacionadas con la energía fotovoltaica, sino también termosolar, eólica, geotermal o biomasa, entre otras.

## 6. solaR

solaR es un paquete de código para el entorno de R, desarrollado por Oscar Perpiñán, que a pesar de tener objetivos diferentes a los planteados por la aplicación desarrollada en este proyecto, ha servido como base para validar todos los cálculos que se han realizado en ella.

En el apartado 2.2 se lleva a cabo un desarrollo mas detallado de las características de las soluciones mencionadas así como sus diferencias con la propuesta de este proyecto.

## 1.3. Aspectos técnicos

Para construir cualquier página web se deben desarrollar dos sistemas diferentes llamados Servidor (conocido también como Backend) y Cliente(también llamado Frontend). El Backend es el la parte de cálculo y tratamiento de peticiones. Por otro lado se encuentra el Frontend que es la parte de cara al usuario, la que se encarga de recoger los datos introducidos por este y enviarlos al servidor.

### 1.3.1. Backend

Para el Backend se ha empleado una tecnología basada en Javascript llamada NodeJS<sup>3</sup>, con la ayuda de las librerías ExpressJS<sup>4</sup> y Mongoose<sup>5</sup>.

La base de datos que se ha utilizado para almacenar los datos necesarios ha sido MongoDB. En el servidor se realizan varias tareas relacionadas con los cálculos necesarios. Algunas de estas tareas son:

- Obtención de los datos de irradiación global media en el plano horizontal para el emplazamiento indicado
- Proceso completo de cálculo para pasar de la irradiación en el plano horizontal al plano inclinado y orientado según los datos introducidos por el usuario
- Proceso de obtención de los datos relacionados con el perfil horario de temperatura en el emplazamiento indicado
- Gestión de las diferentes rutas que constituyen la API.

<sup>3</sup>NodeJS: Entorno de ejecución basado en el motor de Chrome llamado V8. <https://nodejs.org/en/>

<sup>4</sup>ExpressJS: Framework web para el entorno de NodeJS. <https://expressjs.com/es/>

<sup>5</sup>Mongoose: Capa intermedia de interacción para las BBDD MongoDB <https://mongoosejs.com/>

### **1.3.2. Frontend**

Para el Frontend de la página se han utilizado las tres tecnologías necesarias para poder desarrollar una pagina web: HTML5, CSS3, Javascript.

Esta parte de la página es la encargada de recoger los datos del usuario y enviarlos al servidor para que se realicen los cálculos. Una vez realizados dichos cálculos, la página mostrará la información relevante al usuario, junto con algunos unos gráficos adicionales.

Tanto el backend como el frontend están almacenados en un servidor de Linux remoto activo 24/7.

Todas las tareas mencionadas tanto en la parte de Backend como en la parte de Frontend se describirán en detalle en la sección 3, junto con todos los cálculos en los que se ha basado.

## **Capítulo 2**

# **Estado del arte**

### **2.1. Situación actual de la generación fotovoltaica**

Según el informe anual de la UNEF<sup>1</sup> en 2019[1] la evolución de las energías renovables superó incluso las expectativas más optimistas alcanzando valores próximos a los 100 GW.

Los aspectos que más han influido en estas cifras han sido, entre otros, la reducción drástica del coste de producción de dichas tecnologías. De hecho, la energía fotovoltaica es ya más barata que la generada por plantas de combustibles fósiles en términos de LCOE<sup>2</sup>. Según menciona Bloomberg Energy Finance, la fotovoltaica seguirá reduciendo sus costes un 34 % hasta 2030.

Sumado a la reducción del coste, el aumento de compraventa de energía a largo plazo - que sigue en alza desde 2018, alcanzando los 14 GW - es otro de los factores influyentes en el crecimiento del sector. Así lo es también la reducción del precio de la energía en las subastas, alcanzando valores tan bajos como 20\$/MWh.

Concretamente en Europa, el crecimiento anual de la capacidad solar instalada ha sido de un 23 %, con Alemania como líder sumando otros 2,95 GW respecto a la capacidad del año anterior. En segundo y tercer lugar se encuentran Turquía y los Países Bajos.

Bajando un nivel más, nos encontramos con el mercado español, que, según estimaciones del mismo informe de la UNEF, la potencia total instalada experimentó un crecimiento significativo, llegando hasta el valor de 262 MW, sumando la potencia instalada tanto de generación centralizada como la de autoconsumo.

Para este año 2020 se estima que la instalación de energía fotovoltaica alcance el umbral de los 20 GW. Si se cumplen las expectativas, la capacidad podría llegar a alcanzar los 200 GW para 2023.

---

<sup>1</sup>UNEF: Unión Española fotovoltaica

<sup>2</sup>LCOE: Levelized Cost of energy: Medición del coste medio de generación de energía de una planta a lo largo de su vida útil

Un papel importante lo juegan las autoridades tanto nacionales como a nivel europeo, que apuestan por las fuentes de energías limpias. Para ser exactos, el año 2018 fue uno de los más relevantes en materia de política energética europea desde que se aprobó el tercer paquete de energía en 2009. De las ocho propuestas que se aprobaron, destaca por su importancia para el sector fotovoltaico la directiva 2018/2001, en la que se recoge el derecho básico al autoconsumo, individual o colectivo, al almacenamiento y sobretodo a la venta de excedentes.

En el panorama español, tras varios años de parálisis debida a la compleja situación política de los últimos años, la energía fotovoltaica volvió a recuperar algo de impulso a final del año 2019. Durante el año 2018 y especialmente el 2019, se han incrementado sustancialmente las instalaciones de fotovoltaica, en gran parte gracias a las expectativas generadas por la eliminación del denominado impuesto al sol y la progresiva suspensión de trabas administrativas todas ellas propiciadas por el nuevo marco legal establecido por la administración pública mediante los Reales Decretos 15/2019 y 244/2019.

Según datos proporcionados por la Red Eléctrica Española<sup>3</sup>, la potencia instalada en España ha experimentado un crecimiento de 1,2GW en los últimos 4 años, pasando de 4.6 GW en 2016 a 5.8 GW en 2019.

Se trata del mayor ritmo de crecimiento desde 2008, cuando se instalaron cerca de 2.7 GW de nueva potencia. Es una buena noticia, pero no exenta de dificultades debidas sobre todo a la unidireccionalidad de la red eléctrica de nuestro territorio. Sin ir más lejos, el operador técnico del sistema, REE ha denegado la instalación de 26,3 GW de nueva potencia debido a la imposibilidad de los nudos para gestionar la energía producida por dicha capacidad.

Esto no significa que dicha energía no se vaya a instalar, sino que habrá que esperar a la nueva planificación energética prevista para el periodo 2021-2026 en la que ya están trabajando tanto la REE como las comunidades autónomas para brindar más oportunidades para los sistemas de conexión a red.

En definitiva, como podemos observar, las energías renovables, y en especial la fotovoltaica están dando mucho que hablar y cada vez es un tema más tratado por el público general y por tanto, es un aspecto a tener en cuenta a la hora de construir nuevas edificaciones o mejorar la eficiencia energética de las existentes.

Surge por tanto la necesidad de aplicaciones y soluciones para la estimación de instalaciones fotovoltaicas que sean intuitivas y fáciles de usar para aprovechar el gran interés que está mostrando el público general.

## 2.2. Soluciones existentes y sus carencias

Como ya se ha mencionado en el apartado 1.2 existen multitud soluciones para la estimación o simulación de instalaciones fotovoltaicas. Sin embargo, es posible que la aplicación descrita en este proyecto siga teniendo cabida porque ofrece ciertas funciones que, en cierta medida, no existen en las soluciones que se han estudiado.

A continuación se describirán con detalle las soluciones mencionadas anteriormente, junto con sus capacidades y carencias.

<sup>3</sup><https://www.ree.es/es/datos/publicaciones/series-estadísticas-nacionales>

### 1. PVsyst - Photovoltaic Software [2]

El software PVsyst, desarrollado por la empresa suiza con el mismo nombre es quizá el más conocido dentro del ámbito del estudio y la estimación de instalaciones fotovoltaicas. Ofrece una amplia capacidad de personalización de todos los componentes de la instalación.

PVsyst es capaz de calcular con amplio detalle el diseño y dimensionado del sistema, zonas de sombra, envejecimiento del material, almacenamiento de energía, entre otras opciones.

PVsyst se diferencia de la aplicación que se desarrolla en este proyecto en algunos puntos importantes como:

- Es de pago, con una licencia anual de aproximadamente €568.
- Es un programa que se debe instalar en un ordenador de Windows (No funciona en Linux u OSX).
- Es poco intuitivo para un usuario con bajos conocimientos de instalaciones fotovoltaicas.

En resumen, PVsyst es un software mucho mas completo y complejo que la solución que la desarrollada en este proyecto, y que va enfocada a un público con una base sólida sobre energía fotovoltaica.

### 2. CalculationSolar.com [3]

Como ya se ha mencionado en la introducción, CalculationSolar.com es el primer resultado que aparece en el motor de búsqueda de Google cuando se introduce el término “*calculadora de instalaciones fotovoltaicas*” y por tanto será una de las primeras opciones que un usuario que está interesado en realizar una instalación fotovoltaica considere.

A diferencia de PVsyst, CalculationSolar.com es una solución en la web y gratuita, por lo que la barrera de acceso es mas baja. Nada mas entrar a la pagina lo primero con lo que nos encontramos es con un formulario sencillo sobre el emplazamiento y la configuración de la instalación. La información que se solicita es sencilla e intuitiva, incluso se ofrece la posibilidad de hacer clic en un mapa interactivo para determinar las coordenadas.

Lo siguiente es introducir la información acerca de las necesidades de potencia, dado que esta calculadora esta enfocada a las instalaciones de autoconsumo sin conexión a red.

Una vez introducidos estos datos, el programa realiza los cálculos y nos ofrece un resultado bastante detallado del campo fotovoltaico, el regulador de carga, la batería y el inversor que mejor encajaría con nuestro requisitos.

La principal diferencia con la aplicación que se desarrolla en este proyecto es requisitos versus limitaciones. Es decir, CalculationSolar.com te permite seleccionar tus requisitos de potencia y te indica el generador que vas a necesitar para poder hacer frente a dicha carga. En cambio, SolarCalc indica la potencia y energía que se puede llegar a generar con las limitaciones arquitectónicas impuestas por el usuario.

### 3. SISIFO [4]

La solución propuesta por el Grupo de Energías Fotovoltaicas del Instituto de Energía Solar de la Universidad Politécnica de Madrid es también un solución web para el calculo de instalaciones solares tanto de bombeo de agua como conectadas a red.

El sistema de input de información del usuario es muy completo. Paso por paso le permite a éste introducir desde los datos geográficos y meteorológicos hasta hasta los valores de

perdidas en los diferentes cableados del sistema. No obstante, los valores por defecto son muy válidos así que incluso un usuario con poco conocimiento sobre energía podría llegar a obtener unos resultados fiables.

Una vez realizada la simulación, la aplicación aporta multitud de resultados detallados tales como Irradiaciones en el plano horizontal e inclinado, temperaturas y energía producida.

Esta aplicación es la más parecida, salvando las distancias, a la que se desarrolla en este proyecto. Sin embargo, es posible que puede llegar a abrumar en cierta medida a un usuario que desea solamente conocer cual es la potencia o energía que puede llegar a producir en su casa, para saber si le va a resultar rentable la inversión.

#### 4. PVGIS [5]

PVGIS es la solución desarrollada por el JRC (Centro de Investigación Conjunta), que forma parte del EU Science Hub. Consiste en una aplicación web que nos permite estimar la energía que puede llegar a producir una instalación fotovoltaica en función de los parámetros introducidos por el usuario.

Tras elegir la localización como primer paso, el programa nos abre la posibilidad de elegir el tipo de instalación, entre conectada a red, con seguimiento, o autónomo.

En este caso, para poder compararlo con la aplicación de este proyecto, se va a realizar la estimación de un sistema conectado a red. La principal diferencia se presenta cuando el programa solicita al usuario la potencia FV pico instalada, así como la tecnología y las perdidas porcentuales del sistema.

Una vez introducidos estos datos, la aplicación nos ofrece los resultados de producción de energía mensual del sistema.

A diferencia de esto, SolarCalc, solicita al usuario el dato de la superficie disponible, para estimar la potencia máxima y por consiguiente, la energía máxima.

#### 5. System Advisor Model [6]

Esta aplicación desarrollada por el Laboratorio de Energías Renovables del Departamento de Energía de los Estados Unidos. Es un conjunto de soluciones enfocadas a facilitar la toma de decisiones relacionadas con el campo de las Energías Renovables. Entre sus funcionalidades se incluyen programas de cálculo de Fotovoltaica, Termosolar, Eólica, Geotermal o Biomasa.

Es una herramienta muy completa a la vez que compleja, con un enfoque centrado en el apartado económico y la rentabilidad. En cuanto al cálculo y la estimación de una instalación fotovoltaica, ofrece una amplia posibilidad de personalización de cada uno de los campos que afecta a dicho cálculo. Cada uno de los parámetros de la radiación, el módulo, el inversor, las sombras e incluso de la inversión y amortización de la instalación son totalmente ajustables.

Con ésta aplicación sucede como con alguna de las mencionadas anteriormente, que puede llevar a confusión a un usuario medio sin conocimiento relacionados con la fotovoltaica. Estos usuarios son el público objetivo de la aplicación de SolarCalc, desarrollada en este proyecto.

#### 6. solaR [7]

Este paquete para R, desarrollado por Oscar Perpiñán, permite llevar a cabo estudios tanto del rendimiento de los sistemas fotovoltaicos como de la radiación solar. Incluye una serie de clases, métodos y funciones para calcular aspectos como la geometría solar, radiación solar

incidente sobre un generador, realizar el paso de un generador horizontal a un generador inclinado y orientado y simular el rendimiento de diferentes aplicaciones de la energía fotovoltaica.

A pesar de no ser una aplicación de estimación de instalaciones solares como tal, ha sido, junto con el libro [8], la base de contraste y referencia de los cálculos que se han sido necesario llevar a cabo para poder desarrollar la aplicación de SolarCalc.

## Capítulo 3

# Parte teórica y desarrollo del código

El proceso de cálculo que se va a seguir para la estimación completa de la instalación fotovoltaica conectada a red es el que se detalla en el libro de Oscar Perpiñán, tutor de este trabajo, denominado Energía Solar Fotovoltaica [8]. También se harán menciones a las presentaciones que se encuentran en el mismo enlace que el libro.

A lo largo de este capítulo, se utilizará el término **aplicación** para referirse a todo el conjunto de código relacionado tanto con proceso de obtención de todos los datos necesarios como el de mostrar la información relevante al usuario.

A continuación se muestra un diagrama que resume los pasos del proceso que se ha seguido para llegar a los resultados deseados.

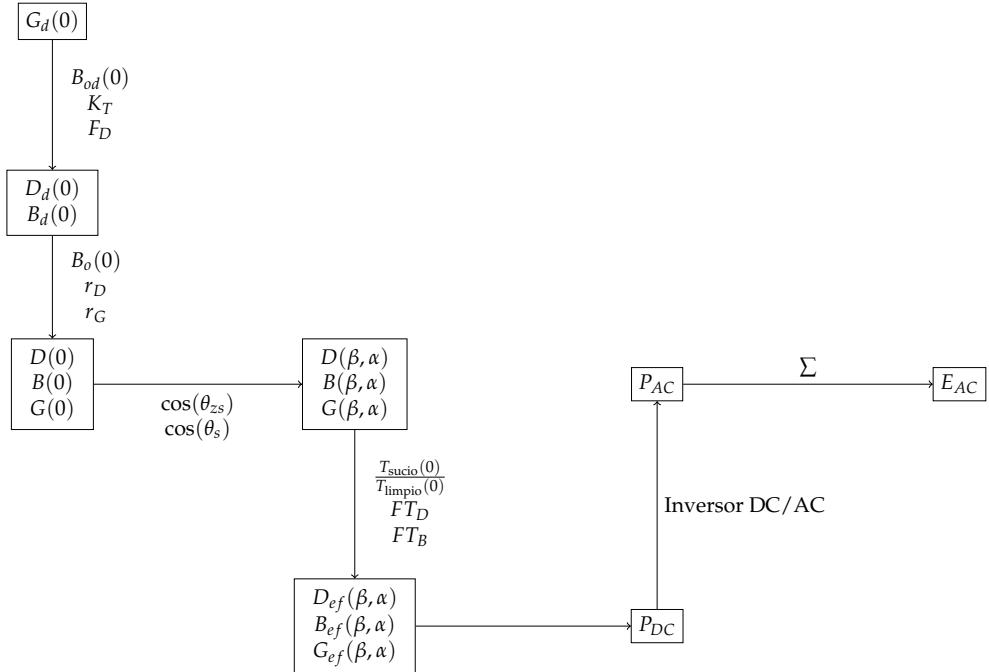


Figura 3.1: Procedimiento de cálculo

Los pasos que se muestran en el diagrama son:

1. Obtener la irradiación global diaria en el plano horizontal en el emplazamiento del usuario
2. Separar dicha irradiación en sus componentes de irradiación difusa y directa.
3. Convertir la irradiación diaria a un perfil horario de irradiación global, directa y difusa en en plano horizontal.
4. Pasar del perfil horario en el plano horizontal a un perfil con la inclinación y orientación indicada por el usuario.
5. Aplicar las pérdidas relacionadas con el ángulo de incidencia y el nivel de suciedad.
6. Calcular la potencia en corriente continua entregada por el generador.
7. Convertir la potencia en corriente alterna a través de inversor.
8. Realizar un sumatorio de las potencias en corriente alterna para obtener la energía entregada.

### 3.1. Naturaleza de la radiación solar

Para el cálculo de la radiación solar que finalmente incide en una área concreta localizada en la superficie terrestre debemos distinguir tres componentes diferenciados, comúnmente denominados:

- **Radiación Directa,  $B$ :** representa la porción de radiación procedente en linea directa del Sol.
- **Radiación Difusa,  $D$ :** representa toda la fracción de radiación procedente de todo el cielo, excepto del sol. Es decir, incluye todos aquellos rayos dispersados por la atmósfera. Por tanto, será dependiente de condiciones climatológicas como nubosidades, niebla o lluvia.
- **Radiación del albedo,  $R$ :** es aquella fracción de la radiación procedente de la reflexión con el suelo. Habitualmente, supone una contribución muy pequeña, que en algunos casos, como el de este proyecto, puede ser despreciada. Normalmente se toma en consideración en áreas con altos índices de reflexión como aquellos con temporadas largas de nieve.

La suma de las tres componentes constituye la denominada radiación global:

$$G = B + D + R \quad (3.1)$$

El Capítulo 3: Radiación solar, del libro mencionado anteriormente [8], describe el proceso que se ha de seguir para obtener una estimación de las componentes directa y difusa a partir del dato de radiación global, dado que es el que comúnmente se puede obtener de una localización determinada.

### 3.1.1. Radiación fuera de la atmósfera terrestre

Lo primero que se menciona en dicho proceso es la obtención de la irradiancia denominada extra-terrestre o extra-atmosférica, que es la radiación que llega a la atmósfera, directamente desde el sol, que no sufre ninguna perdida por interaccionar con algún medio. Como la relación entre el tamaño de nuestro planeta y la distancia entre el Sol y la Tierra es muy reducida, es posible asumir que el valor de dicha irradiancia es constante, siendo este valor  $B_0 = 1367 \frac{W}{m^2}$ , según varias mediciones llevadas a cabo.

Como la órbita que describe la Tierra alrededor del Sol no es totalmente circular, sino que tiene cierta excentricidad, a la hora de calcular la irradiancia incidente en una superficie tangente a la atmósfera en una latitud concreta, debemos aplicar un factor de corrección de la excentricidad, como se muestra en la ecuación:

$$B_0(0) = B_0 \epsilon_0 \cos \theta_{zs} \quad (3.2)$$

Para calcular esta irradiación diaria, debemos calcular primero los diferentes componentes que forman parte de esta:

- $B_0 = 1367 \frac{W}{m^2}$
- Factor de corrección por excentricidad:  $\epsilon_0 = 1 + 0,033 \cdot \cos(2\pi d_n / 365)$

En código se representa de la siguiente manera:

```
1 const exct = 1 + 0.033 * Math.cos((2 * Math.PI * elem.normalDay) / 365)
```

Extracto de código 3.1: Factor de corrección por excentricidad

- Ecuación de Cooper para la declinación:  $\delta = 23,45 \text{ deg } \sin\left(\frac{2\pi(d_n+284)}{365}\right)$

En código se representa de la siguiente manera:

```
1 const decl = 23.45 * Math.sin((2 * Math.PI * (elem.normalDay + 284)) / 365)
```

Extracto de código 3.2: Ecuación de Cooper para declinación

- Cenit Solar:  $\cos(\theta_{zs}) = \cos(\delta) \cos(\omega) \cos(\phi) + \sin(\delta) \sin(\phi)$

Para el calculo del cenit solar, hay que tener en cuenta que varía en función de la hora del día, así que se deberá calcular un valor para cada una de las 24h del día, con la diferencia de que en el cálculo, las horas irán de -12 a 12 en lugar de 0 a 23 y además, tendremos que convertir las horas a ángulos multiplicando por  $15^\circ$ . De tal manera, que en código, el calculo del cenit solar, tendrá ésta representación:

```

1   for (let h = -12; h < 12; h++) {
2     const cosZenit =
3       Math.cos(deg2rad(elem.decl)) * Math.cos(deg2rad(h * 15)) *
4       Math.cos(deg2rad(newData.latitude)) +
5       Math.sin(deg2rad(elem.decl)) * Math.sin(deg2rad(newData.latitude))
6     const zenithVal = rad2deg(Math.acos(cosZenit))
7   }

```

Extracto de código 3.3: Cálculo del cenit solar

Como se puede observar, la irradiancia extra-terrestre solo requiere de consideraciones geométricas. Así, integrando la ecuación 3.2, podemos calcular la irradiación diaria extra-terrestre con la ecuación obtenida:

$$B_{0d}(0) = -\frac{T}{\pi} B_0 \epsilon_0 (\omega_s \sin \phi \sin \delta + \cos \phi \cos \delta \sin \omega_s) \quad (3.3)$$

Se puede llegar a demostrar que la media mensual de la irradiación diaria coincide en valor con la irradiación diaria de un día denominado día promedio. Por tanto, podemos calcular el valor medio mensual de la irradiación diaria extra-atmosférica con el valor de la declinación de uno de los doce días promedio.

Estos doce días promedios son:

Mes	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic
$d_n$	17	45	74	105	135	161	199	230	261	292	322	347

Cuadro 3.1: Días promedio

Aplicando estos cálculos a la ecuación 3.3 podemos calcular el valor de la irradiancia extra-terrestre diaria  $B_{0d}(0)$ :

```

1   const B0d0 =
2     -(24 / Math.PI) *
3     B0 *
4     elem.exct *
5     (elem.ws * Math.sin(deg2rad(newData.latitude)) * Math.sin(deg2rad(elem.decl)) +
6      Math.cos(deg2rad(elem.decl)) * Math.cos(deg2rad(newData.latitude)) *
7      Math.sin(elem.ws))

```

Extracto de código 3.4: Ecuación para B0d0

donde:

- $B_0$ : valor constante de  $1367 \text{ W/m}^2$
- $\text{exct}$ : excentricidad
- $\omega_s$ : ángulo amanecer
- $\text{decl}$ : declinación

### 3.1.2. Cálculo de componentes de radiación solar

Para poder calcular la energía producida por un generador fotovoltaico, se necesita conocer la radiación solar que incide sobre la superficie de dicho generador. En éste caso las características del generador vienen indicadas por el usuario de la aplicación, por tanto, debemos recoger dicha información. La información necesaria para conocer la radiación que incide en la superficie son:

- **Latitud y longitud del emplazamiento:** La ruta que se ha tomado para obtener estos datos, es la de pedirle al usuario su dirección, o una dirección cercana a su localización, y utilizar la API de Google Maps<sup>1</sup> para convertir dicha dirección en las coordenadas de latitud y longitud que se necesitan para poder extraer la irradiación mencionada anteriormente.
- **Orientación, inclinación y nivel de suciedad de la superficie:** Estos valores son recogidos directamente de los campos de la página web, sin necesitar ningún trato especial.

En el anexo de programación se explica más detalladamente como se ha realizado la recogida de estos datos y su posterior envío a la API de Google.

### Obtención de la irradiación global media en el plano horizontal

Como se puede observar en el diagrama 3.1, el primer paso del cálculo de una instalación fotovoltaica es el de conocer la irradiación global media en el plano horizontal para el emplazamiento donde se va a realizar el cálculo.

Anteriormente se obtuvieron las coordenadas de latitud y longitud del emplazamiento introducido por el usuario.

A continuación, se va a obtener la irradiación para las dichas coordenadas utilizando un servicio de ADRASE<sup>2</sup>, un proyecto realizado por el CIEMAT. Este servicio ofrece de manera gratuita y de uso libre, datos correspondientes a valores medios mensuales de irradiación global en el plano horizontal para toda la geografía española.

Los datos están disponibles a través de un mapa interactivo y de unos enlaces personalizados que incluyen la valores de latitud y longitud para los que se desea obtener dicha información. En esos enlaces se ofrecen los datos de irradiación global mínima, media y máxima en el plano horizontal. Con el objetivo de no saturar la página de ADRASE y evitar errores de funcionamiento de la aplicación debidos a la posibilidad de que la página desaparezca en un futuro, se ha realizado una descarga de los datos de la página, con un intervalo de  $0.1^\circ$  tanto en longitud como en

---

<sup>1</sup>API Google Maps: Enlace interactivo al que se le pueden enviar los datos de una dirección y devuelve las coordenadas de latitud y longitud de un emplazamiento.

<sup>2</sup>ADRASE: Acceso a Datos de Irradiación Solar en España <http://www.adrase.com/>

latitud y se han guardado en una base de datos. De esta forma también se reducen los tiempos de cálculo en gran medida al tener un acceso casi instantáneo a los valores medios de irradiación. Una explicación detallada de como se ha realizado dicha descarga se incluye en el la sección 5.2.3 del Anexo I.

Una vez obtenida la radiación global media para los doce meses, en el emplazamiento indicado, se puede comenzar el proceso de obtención de las componentes de ésta.

### Obtención de las componentes de radiación global

En 1960, Liu y Jordan [9] expusieron una forma de caracterizar la radiación solar en un lugar, mediante el índice de claridad  $K_T$ . Éste índice relaciona la radiación global y la extra-terrestre, ambas en el plano horizontal. La expresión de el indice de claridad diario es:

$$K_{Td} = \frac{G_d(0)}{B_{0d}(0)} \quad (3.4)$$

mientras que el indice de claridad mensual es la relación entre las medias mensuales de irradiación diaria:

$$K_{Tm} = \frac{G_{d,m}(0)}{B_{0d,m}(0)} \quad (3.5)$$

En el código, el índice de claridad media se calcula con la siguiente expresión:

```

1 newData.meanValues.forEach((elem, index) => {
2     const Ktd = elem.meanGR / elem.B0d0
3 })

```

Extracto de código 3.5: Índice de claridad diario

Para cada uno de los valores medios mensuales, dividimos el valor de radiación global media entre el  $B_{0d}(0)$  calculado anteriormente.

Habiendo calculado el índice de claridad, podemos utilizar su valor para calcular la Fracción de radiación difusa en el plano horizontal ( $F_D = \frac{D(0)}{G(0)}$ ). Para ello, podemos utilizar la ecuación de Page para aproximar el dicho valor conociendo el índice de claridad.

La ecuación que define éste valor es:

$$F_{Dm} = 1 - 1,13K_{Tm} \quad (3.6)$$

En código, la expresión es:

```

1 newData.meanValues.forEach((elem, index) => {
2     const Fd = 1 - 1.13 * elem.Ktd
3 })

```

Extracto de código 3.6: Fracción de difusa

Donde elem.Ktd es el índice de claridad calculado en el paso anterior.

Al conocer, mediante la ecuación de Page el valor de la fracción de radiación difusa, podemos aplicar dicha relación para calcular los valores de radiación directa y difusa en el plano horizontal, a partir de la radiación global. Las dos ecuaciones necesarias para calcular estos valores son:

$$D_d(0) = F_D G_d(0) \quad (3.7)$$

$$B_d(0) = G_d(0) - D_d(0) \quad (3.8)$$

En código, las expresiones toma la siguiente forma:

```

1  newData.meanValues.forEach((elem, index) => {
2      const Dd0 = elem.Fd * elem.meanGR
3      const Bd0 = elem.meanGR - Dd0
4  })

```

Extracto de código 3.7: Radiación directa y difusa

Para cada uno de los doce valores de radiación global media, aplicamos la fracción calculada anteriormente para obtener los dos valores de radiación en el plano horizontal.

## 3.2. Radiación en superficies inclinadas

Habiendo ya obtenido las valores de irradiación diaria en el plano horizontal, y divididos en sus componentes directa y difusa. A continuación, para poder llevar a cabo las transformaciones al plano inclinado, debemos estimar los valores horarios de irradiancia difusa, directa y global en el plano horizontal. Con estas estimaciones podemos calcular los valores correspondientes al plano inclinado.

Integrando los valores de irradiancia se obtienen las estimaciones de irradiación diaria difusa, directa y global en el plano inclinado. Al realizar ésta transformación, la irradiancia/irradiación recibe el adjetivo de *incidente*.

A estos valores les aplicaremos las pérdidas por suciedad, añadiéndole el adjetivo de *efectiva* a la irradiancia e irradiación. Será esta irradiación incidente efectiva la que se utilizará para el cálculo de la energía producida por el generador.

### 3.2.1. Estimación de irradiancia a partir de irradiación diaria

Partiendo de los valores calculados anteriormente de irradiación diaria difusa, directa y global en el plano horizontal, podemos realizar la transformación al plano inclinado estimando el perfil de irradiancia correspondiente a cada valor de irradiación.

Como la variación solar durante una hora es relativamente baja, podemos suponer que el valor medio de irradiancia durante esa hora coincide numéricamente con irradiación horaria ( $\frac{kWh}{m^2}$ ).

Por otro lado, mediante análisis de series temporales se ha demostrado que la relación entre la irradiancia y la irradiación difusa es equivalente a la existente entre la irradiancia y la irradiación extra-atmosférica.

$$r_D = \frac{D(0)}{D_d(0)} = \frac{B_0(0)}{B_{0d}(0)} \quad (3.9)$$

El factor  $r_D$  se puede calcular mediante la siguiente expresión:

$$r_D = \frac{\pi}{T} \cdot \frac{\cos(\omega) - \cos(\omega_s)}{\omega_s \cdot \cos(\omega_s) - \sin(\omega_s)} \quad (3.10)$$

Donde:

- T: Duración del día en horas
- $\omega_s$ : ángulo de amanecer, expresado en radianes
- $\omega$ : instante central de la hora correspondiente

La implementación en código es la siguiente

```

1 const rd = []
2 for (let h = -12; h < 12; h++) {
3     const hRad = Math.cos(deg2rad(h * 15))
4     const rdval = (Math.PI / 24) * ((hRad - Math.cos(elem.ws)) / (elem.ws *
5         Math.cos(elem.ws) - Math.sin(elem.ws)))
6     rd.push(rdval)
7 }
```

#### Extracto de código 3.8: Cálculo de rD

Para cada uno de los valores horarios de -12 a 12, se calcula el angulo horario multiplicando por 15 grados y se introduce en la ecuación, obteniendo 24 valores por cada uno de los meses.

El mismo análisis muestra también la relación entre la irradiancia e irradiación global asimilable a una función dependiente de la hora solar.

$$r_G = \frac{G(0)}{G_d(0)} r_D \cdot (a + b(\omega)) \quad (3.11)$$

siendo:

$$a = 0,409 - 0,5016 \cdot \sin(\omega_s + \frac{\pi}{3}) \quad (3.12)$$

$$a = 0,6609 - 0,4767 \cdot \sin(\omega_s + \frac{\pi}{3}) \quad (3.13)$$

donde  $\omega_s$  es negativa y está expresada en radianes.

En código, los expresiones anteriores se representan de la siguiente forma:

```

1 const rg = []
2 const a = 0.409 - 0.5016 * Math.sin(elem.ws + Math.PI / 3)
3 const b = 0.6609 + 0.4767 * Math.sin(elem.ws + Math.PI / 3)
4 let i = 0
5 for (let h = -12; h < 12; h++) {
6     const hRad = Math.cos(deg2rad(h * 15))
7     const rgval = rd[i] * (a + b * hRad)
```

```

8     rg.push(rgval)
9     i++
10    }

```

Extracto de código 3.9: Cálculo de rG

### 3.2.2. Transformación al plano del generador

Para obtener los valores de irradiancia en el plano inclinado, lo primero que debemos calcular son los valores horarios de radiación global, directa y difusa en el plano horizontal, aplicando las dos relaciones,  $r_D$  y  $r_G$  que se han calculado en la sección anterior.

Para ello, emplearemos las siguientes ecuaciones:

$$D_h = r_D \cdot D_d(0) \quad (3.14)$$

$$G_h = r_G \cdot G(0) \quad (3.15)$$

$$B_h = G_h - D_h \quad (3.16)$$

En código, las expresiones serían las siguientes:

```

1  const hourlyValues = []
2  let hour = -12
3  const dawn = rad2deg(elem.ws) / 15
4  elem.dawn = dawn
5  for (let i = 0; i < 24; i++) {
6      const cosHour = Math.cos(deg2rad(hour * 15))
7      const cosWs = Math.cos(elem.ws)
8      if (cosHour > cosWs) {
9          const Dh = elem.rd[i] * elem.Dd0
10         const Gh = elem.rg[i] * elem.meanGR
11         const Bh = Gh - Dh
12         hourlyValues.push({
13             Bh, Dh, Gh, cosHour, cosWs,
14         })
15     } else {
16         hourlyValues.push({
17             Bh: 0, Dh: 0, Gh: 0, cosHour, cosWs,
18         })
19     }
20  hour++

```

```

21     }
22     elem.hourlyValues = hourlyValues

```

### Extracto de código 3.10: Cálculo de rG

Se puede observar que para las horas en las que el coseno es menor que el coseno del ángulo de amanecer, se toman valores nulos para las radiaciones. Dado que en esas horas no existe radiación solar, aunque en algunos casos el cálculo pueda dar valores distintos de 0.

Una vez obtenidos estos valores horarios de irradiancia en el plano horizontal, el siguiente paso consiste en llevarlo al plano del generador.

El planteamiento de ésta transformación es el siguiente:

- **Irradiancia Directa**  $B(\beta, \alpha)$ : ecuación basada en geometría solar (ángulo cenital) y del generador (ángulo de incidencia).

$$B(\beta, \alpha) = B(0) \cdot \frac{\max(0, \cos(\theta_s))}{\cos(\theta_{zs})} \quad (3.17)$$

- **Irradiancia Difusa**  $D(\beta, \alpha)$ : Utilizando el modelo de cielo anisotrópico (Pág 31 [8]), se distinguen dos componentes de la irradiancia difusa, denominados *circunsolar* e *isotrópica*.

$$D(\beta, \alpha) = D^I(\beta, \alpha) + D^C(\beta, \alpha) \quad (3.18)$$

$$D^I(\beta, \alpha) = D(0) \cdot (1 - k_1) \cdot \frac{1 + \cos(\beta)}{2} \quad (3.19)$$

$$D^C(\beta, \alpha) = D(0) \cdot k_1 \cdot \frac{\max(0, \cos(\theta_s))}{\cos(\theta_{zs})} \quad (3.20)$$

$$k_1 = \frac{B_h}{B_0(0)} \quad (3.21)$$

- **Irradiancia de Albedo**  $R(\beta, \alpha)$ : Se considera nula por tener un porcentaje de aportación muy reducido.

Estos componentes sumados, forman la expresión de la irradiancia global en el plano del generador.

$$G(\beta, \alpha) = D(\beta, \alpha) + B(\beta, \alpha) + R(\beta, \alpha) \quad (3.22)$$

### Ángulo de incidencia

En la ecuación de la irradiancia directa, uno de los elementos que influyen en el cálculo es el angulo de incidencia (Pág. 14 [8]), cuya expresión es:

$$\begin{aligned}
\cos(\theta_s) = & \text{sign}(\phi) \cdot [\sin(\beta) \cos(\alpha) \cos(\delta) \sin(\phi) - \\
& - \sin(\beta) \cos(\alpha) \cos(\phi) \sin(\delta)] + \\
& + \sin(\beta) \sin(\alpha) \cos(\delta) \sin(\omega) + \\
& + \cos(\beta) \cos(\delta) \cos(\omega) \cos(\phi) + \\
& + \cos(\beta) \sin(\delta) \sin(\phi)
\end{aligned} \quad (3.23)$$

Donde:

- $\beta$ : Inclinación del generador
- $\alpha$ : Orientación del generador
- $\phi$ : Latitud del emplazamiento
- $\delta$ : Declinación
- $\omega$ : Hora solar

Traducida a código, la expresión será:

```

1   for (let h = -12; h < 12; h++) {
2       const betaRad = deg2rad(newData.angle)
3       const alphaRad = deg2rad(newData.orientation)
4       const hRad = deg2rad(h * 15)
5       const latRad = deg2rad(newData.latitude)
6       const declRad = deg2rad(elem.decl)
7       const anglIncidenciaVal =
8           Math.sign(newData.latitude) *
9           (Math.sin(betaRad) * Math.cos(alphaRad) * Math.cos(declRad) *
10            Math.cos(hRad) * Math.sin(latRad) - Math.sin(betaRad) *
11            Math.cos(alphaRad) * Math.cos(latRad) * Math.sin(declRad)) +
12            Math.sin(betaRad) * Math.sin(alphaRad) * Math.cos(declRad) *
13            Math.sin(hRad) + Math.cos(betaRad) * Math.cos(declRad) *
14            Math.cos(hRad) * Math.cos(latRad) + Math.cos(betaRad) *
15            Math.sin(declRad) * Math.sin(latRad)
16       const anglIncidencia = rad2deg(Math.acos(anglIncidenciaVal))
17       const anglIncidencia = rad2deg(Math.acos(anglIncidenciaVal))
18       incid.push(anglIncidencia)
19       elem.incid = incid
20   }

```

Extracto de código 3.11: Cálculo del ángulo de incidencia

## Cálculo de las componentes

Teniendo el valor del ángulo de incidencia, podemos proceder a calcular las dos componentes de irradiancia para el plano del generador.

```

1 newData.meanValues.forEach((elem, index) => {
2     for (let i = 0; i < 24; i++) {
3         const zenithRad = deg2rad(elem.zenit[i])
4         const incidRad = deg2rad(elem.incid[i])
5         const betaRad = deg2rad(newData.angle)
6         const numerator = Math.max(0, Math.cos(incidRad))
7         const denominator = Math.cos(zenithRad)
8         const Btilt = elem.hourlyValues[i].Bh * (numerator / denominator)
9         const k1 = elem.hourlyValues[i].Bh / elem.B00[i]
10        const DcTilt = elem.hourlyValues[i].Dh * k1 * (numerator / denominator)
11        const DiTilt = elem.hourlyValues[i].Dh * (1 - k1) * ((1 + Math.cos(betaRad)) / 2)
12        const Dtilt = DcTilt + DiTilt
13        const Gtilt = Btilt + Dtilt
14        elem.hourlyValues[i] =
15            ...elem.hourlyValues[i],
16            Btilt,Dtilt,DcTilt,DiTilt,Gtilt,cosZenit: Math.cos(zenitRad),
17        }
18    }
19 })

```

Extracto de código 3.12: Cálculo del ángulo de incidencia

Con estos cálculos obtenemos la irradiancia global incidente, es decir, en el plano del generador. El siguiente es añadirle el apellido de *efectiva* aplicándole las pérdidas por ángulo de incidencia y suciedad.

### 3.2.3. Pérdidas por ángulo de incidencia y suciedad

Al tratarse de sistemas estáticos, la radiación que incide en el módulo se puede ver desviada de la perpendicular e incidir con ángulos diferentes. Esta desviación, cuantificada por el ángulo de incidencia [eq. 3.23], es causa de pérdidas por reflexión.

Por otro lado, la suciedad acumulada en la superficie del módulo altera las propiedades angulares del mismo y reduce la capacidad de transmisión del vidrio.

Estos dos fenómenos reducen la irradiancia que es aprovechada por el módulo. Para el caso de la radiación directa, la expresión de irradiancia efectiva es la siguiente:

$$B_{ef}(\beta, \alpha) = B(\beta, \alpha) \cdot \frac{T_{sucio}(0)}{T_{limpio}(0)} \cdot (1 - FT_B(\theta_s)) \quad (3.24)$$

donde  $FT_B(\theta_s)$  es el factor de pérdidas angulares para la irradiancia directa, calculable mediante la ecuación:

$$FT_B(\theta_s) = \frac{\exp(-\frac{\cos(\theta_s)}{a_r}) - \exp(-\frac{1}{a_r})}{1 - \exp(-\frac{1}{a_r})} \quad (3.25)$$

Los valores del coeficiente de pérdidas angulares deben ser determinados de forma experimental. En la tabla quedan recogidos algunos valores característicos de un módulo de silicio monocristalino convencional para los diferentes grados de suciedad. Además en la tabla también se recogen los valores de la transmitancia al interior del módulo en incidencia normal respecto a uno limpio  $\frac{T_{sucio}(0)}{T_{limpio}(0)}$ .

Grado de suciedad	$\frac{T_{sucio}(0)}{T_{limpio}(0)}$	$a_r$	$c_2$
Limpio	1	0,17	-0.069
Bajo	0,98	0,20	-0.054
Medio	0,97	0,21	-0.049
Alto	0,92	0,22	-0.023

Cuadro 3.2: Coeficiente de pérdidas angulares y transmitancia relativa en incidencia normal para diferentes niveles de suciedad

El cálculo del coeficiente  $FT_B(\theta_s)$ , en código, toma este aspecto:

```

1 const FTB = (Math.exp(-Math.cos(tiltRad) / ar) -
2   Math.exp(-1 / ar)) / (1 - Math.exp(-1 / ar))
3 const Btilt = elem.hourlyValues[i].Btilt * TDirtTClean * (1 - FTB)

```

Extracto de código 3.13: Cálculo del coeficiente  $FT_B(\theta_s)$

A la hora de aplicar dichas pérdidas a la componente difusa, debemos calcular cada una de las dos partes por separado. Para la componente de isotrópica existe otra expresión que depende del ángulo de inclinación del generador, del coeficiente de pérdidas angulares y de dos coeficientes de ajuste  $c_1$  y  $c_2$ . El primero de ellos toma un valor constante  $c_1 = \frac{4}{3\pi}$ . El segundo depende linealmente de  $a_r$  según se recoge en la tabla 3.2.3. En ésta expresión el ángulo  $\beta$  está en radianes.

$$FT_D \simeq \exp\left[-\frac{1}{a_r} \cdot \left(c_1 \cdot \left(\sin \beta + \frac{\pi - \beta - \sin \beta}{1 + \cos \beta}\right) + c_2 \cdot \left(\sin \beta + \frac{\pi - \beta - \sin \beta}{1 + \cos \beta}\right)^2\right)\right] \quad (3.26)$$

Para estas componentes el cálculo de irradiancia efectiva es similar al de la irradiancia directa. Para la componente difusa circunsolar emplearemos el factor de pérdidas angulares de la irradiancia efectiva:

$$D_{ef}^I(\beta, \alpha) = D^I(\beta, \alpha) \cdot \frac{T_{sucio}(0)}{T_{limpio}(0)} \cdot (1 - FT_D(\beta)) \quad (3.27)$$

$$D_{ef}^C(\beta, \alpha) = D^C(\beta, \alpha) \cdot \frac{T_{sucio}(0)}{T_{limpio}(0)} \cdot (1 - FT_B(\beta)) \quad (3.28)$$

En código las ecuaciones toman esta forma:

```

1   const expFTD =
2     -(1 / ar) *
3       (c1 * (Math.sin(betaRad) + (Math.PI - betaRad - Math.sin(betaRad)) / (1 +
4         Math.cos(betaRad))) +
5           c2 * Math.pow(Math.sin(betaRad) + (Math.PI - betaRad -
6             Math.sin(betaRad)) / (1 + Math.cos(betaRad)), 2))
7   const FTD = Math.exp(expFTD)
8
9   const DiTilt = elem.hourlyValues[i].DiTilt * TDirtTClean * (1 - FTD)
10  const DcTilt = elem.hourlyValues[i].DcTilt * TDirtTClean * (1 - FTB)

```

Extracto de código 3.14: Cálculo de las componentes efectivas

Antes de pasar a calcular la energía producida por el generador configurado, se va a llevar a cabo una filtración de datos para eliminar los valores horarios entre el atardecer y el amanecer. Para ello, aplicamos este algoritmo a los datos calculados anteriormente

```

1 newData.meanValues.forEach((elem, index) => {
2   const significantValues = []
3   elem.hourlyValues.forEach((value, index) => {
4     if (Math.abs(value.hour) < Math.abs(elem.dawn)) {
5       significantValues.push(value)
6     }
7   })
8   elem.significantValues = significantValues
9 })

```

Extracto de código 3.15: Selección de los valores clave

Como se observa, los valores horarios menores al amanecer, en valor absoluto, no se introducen en la lista de valores significativos.

### 3.3. Cálculo de la energía producida por el generador

Una vez calculados los valores de radiación en el plano del generador, el siguiente paso será proceder a estimar la energía que es capaz de producir dicho generador conectado a red.

En el caso de la aplicación desarrollada en el proyecto, el objetivo es estimar la energía máxima que será capaz de producir un sistema fotovoltaico conectado a red.

#### 3.3.1. Definición de un SFCR

Un Sistema Fotovoltaico Conectado a Red (SFCR) es un sistema cuya función es producir energía eléctrica en unas condiciones específicas, de tal manera que ésta pueda ser inyectada en la red eléctrica convencional. Para ello, un SFCR se compone del propio generador fotovoltaico, un inversor DC/AC y un conjunto de protecciones eléctricas. Tal y como se muestra en la figura 3.2.

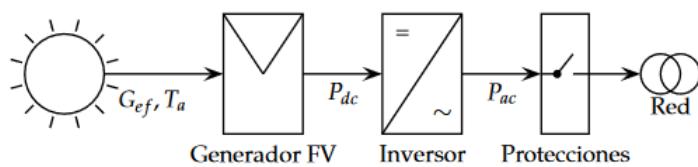


Figura 3.2: Esquema de un SFCR (Figura 6.1 pág 65 [8])

La energía producida por el sistema es consumida total o parcialmente por la unidad doméstica donde está instalada. La energía sobrante es inyectada a la red de distribución, y a cambio el propietario es compensado económicamente a través de sistemas de retribución tales como: la retribución con prima o el balance neto.

A diferencia de un sistema autónomo o de bombeo, el diseño de SFCR no necesita considerar un consumo mínimo a satisfacer. Con este mecanismo, el objetivo es que la producción anual del sistema sea la máxima posible.

#### 3.3.2. Configuración de los elementos del sistema

A la hora de estimar la energía producida por un SFCR, debemos definir las características nominales de los componentes que forman parte de este, como se muestra en la figura 3.2. Con el objetivo de reducir al máximo la complejidad de la aplicación de cara al usuario final (cuyos conocimientos de fotovoltaica, de media, serán reducidos) se han fijado unos valores estándar para el módulo fotovoltaico, la configuración del generador y el inversor.

Estos valores se muestran en las siguientes tablas:

- **Configuración del módulo** El módulo elegido es uno disponible comercialmente, fabricado por JINKO SOLAR, modelo JKM320PP<sup>3</sup>. El área del modulo es de  $1,957 \times 0,922m^2$  y tiene una potencia nominal de  $320W_p$
- **Inversor:**

<sup>3</sup>Módulos solar JINKO SOLAR, modelo JKM320PP <https://autosolar.es/pdf/Ficha-Tecnica-Jinko-Solar-305-320W.pdf>

$G^*$	1000 $\frac{W}{m^2}$
$V_{oc}^*$	46,4 V
$I_{sc}^*$	9,05 A
$V_{mpp}^*$	37,4 V
$I_{mpp}^*$	8,56 A
$N_{cs}$	12
$N_{cp}$	6
TONC	45 °
$T_c$	25 °
$V_t$	0,025V
$m$	1,3

Cuadro 3.3: Configuración módulo standard

$k_0$	0,01
$k_1$	0,025
$k_2$	0,05
Potencia	40 kW
$V_{min}$	420 V
$V_{max}$	750

Cuadro 3.4: Configuración del inversor

- **Generador** La configuración del generador será de **12** módulos en serie y **11** módulos en paralelo.

### 3.3.3. Funcionamiento de una célula solar

Para poder llevar a cabo la estimación de la energía producida por el generador, debemos conocer como influye factores como la radiación o la temperatura en una célula solar y en los valores de tensión y corriente que se alcanzan en dichas condiciones.

Existen cuatro puntos clave que definen una célula solar, la corriente de cortocircuito ( $I_{sc}$ ), la tensión de circuito abierto ( $V_{oc}$ ) y la corriente y tensión en el punto de máxima potencia

#### Punto de máxima potencia

Por otro lado, existe un punto en la curva de funcionamiento del generador conocido como punto de máximo potencia, donde, como su propio nombre indica, los valores de tensión y corriente

son tales que la potencia que está entregando el generador es máxima, y por tanto, es interesante calcular dicho punto.

### Factor de forma y eficiencia

Existe un valor que relaciona los valores más significativos de un módulo fotovoltaico, éste se conoce como factor de forma y su expresión es la siguiente:

$$FF = \frac{I_{mpp} \cdot V_{mpp}}{I_{sc} \cdot V_{oc}} \quad (3.29)$$

Este factor toma valores entre 0,7 y 0,8 variando poco de unas células a otras. Conociendo los valores de  $I_{sc}$  y  $V_{oc}$  es posible calcular la potencia en el punto de máxima potencia, dado que  $P_{mpp} = FF \cdot I_{sc} \cdot V_{oc}$ .

Por otra parte, la calidad de una célula se puede cuantificar según la ecuación de la eficiencia de conversión:

$$\eta = \frac{I_{mpp} \cdot V_{mpp}}{P_L} \quad (3.30)$$

donde  $P_L$  representa la potencia luminosa que incide en la célula.

### Influencia de la temperatura y la radiación

Es importante tener en cuenta, a la hora de entender el funcionamiento de una célula solar, la influencia de los dos principales factores externos: la temperatura ambiente y la iluminación incidente o radiación.

El aumento de la temperatura tiene una influencia notable en la tensión de circuito abierto de la célula según el valor  $dV_{oc}/dT_c$ . donde  $T_c$  es la temperatura de la célula, dependiente de la temperatura ambiente y la radiación incidente. La forma de calcular ésta temperatura depende de las características constructivas del módulo que encapsula a la célula. Si el fabricante no nos ofrece información sobre este valor, para células de silicio cristalino se común tomar el siguiente valor:

$$\frac{dV_{oc}}{dT_c} = -2,3 \frac{mV}{^{\circ}C} \quad (3.31)$$

Sin embargo, el efecto que tiene la irradiancia sobre la tensión de circuito abierto es mucho menor que el de la temperatura. En el caso de éste proyecto, no se tendrá en cuenta dicha influencia.

Sin embargo, como se verá más adelante, en la sección 3.3.4, la radiación afecta a la corriente de cortocircuito, siendo de hecho el factor más significativo en la ecuación.

#### 3.3.4. Comportamiento térmico de un módulo

La mayoría de las ecuaciones que definen el comportamiento de un módulo fotovoltaico se establecen en lo que se conocen como condiciones estándar de funcionamiento. En estas condiciones, la temperatura de la célula es de 25°C. Sin embargo, la temperatura de operación de la célula es diferente y depende directamente de la radiación que recibe el módulo en cada momento.

El módulo recibe una cantidad de radiación dada, absorbiendo la fracción de ésta que no se refleja al exterior. De dicha fracción parte es transformada en energía eléctrica mientras que el resto se entrega en forma de calor al entorno.

Para simplificar podemos asumir que el incremento de la temperatura de la célula respecto a la ambiente depende linealmente de la irradiancia incidente en ésta. El coeficiente de proporcionalidad depende de muchos factores, tales como el modo de instalación del módulo, la velocidad del viento, la humedad del ambiente o las características constructivas del cristal que cubre las células. Todos estos factores quedan recogidos en un valor único representado por la temperatura de operación nominal de la célula (NOCT o TONC), definida como aquella que alcanza una célula cuando su módulo trabaja en las siguientes condiciones:

- Irradiancia:  $G = 800 \frac{W}{m^2}$ .
- Espectro: el correspondiente  $AM = 1,5$ .
- Incidencia normal.
- Temperatura ambiente:  $T_a = 20^\circ C$ .
- Velocidad del viento:  $v_v = 1 \frac{m}{s}$ .

Este valor influye en la temperatura de la célula, como se ha mencionado anteriormente. La expresión que relaciona la temperatura de la célula  $T_c$  con la temperatura de operación nominal TONC es:

$$T_c = T_a + \frac{G_{ef} \cdot (TONC - 20)}{800} \quad (3.32)$$

Como se puede observar, en la temperatura de la célula influyen tanto la temperatura ambiente como la radiación global eficaz. Al trabajar con valores horarios, es necesario conocer el valor de la temperatura ambiente en cada hora del día.

### Cálculo del perfil horario de temperaturas

El procedimiento para calcular los valores de temperatura ambiente horarios para cada uno de los meses en el emplazamiento indicado por el usuario no es trivial pues normalmente no suele haber un registro histórico de temperaturas. Para ello vamos a utilizar el proceso descrito en el documento [10].

En dicho documento se describe un proceso para desarrollar un perfil horario de temperaturas partiendo de una temperatura máxima y mínima.

Los pasos a seguir son:

1. **Obtención de la temperatura mínima y máxima:** Lo primero que necesitamos para poder realizar el perfil de temperatura es conocer los valores de temperatura mínima y máxima para cada uno de los meses. Para ello utilizaremos la API que nos proporciona AEMET a través de su plataforma de OPENDATA<sup>4</sup>. Para ello buscamos en su base de datos la estación climatológica más cercana a las coordenadas del usuario y solicitamos la información necesaria, en este caso las temperaturas mensuales de un año entero.

---

<sup>4</sup>AEMET OPENDATA: Base de datos abierta de información climatológica, entre otras <https://opendata.aemet.es>

**2. Cálculo del ángulo de amanecer:** Una vez obtenidos los valores de temperatura mensuales, el siguiente paso es calcular el ángulo de amanecer para cada uno de los 12 días normales, como se ha hecho en pasos anteriores mediante la declinación. Para ello empleamos las dos ecuaciones:

$$\delta = 23,45 \cdot \sin\left(\frac{2\pi \cdot (d_N + 284)}{365}\right) \quad (3.33)$$

$$\begin{aligned} \cos(\omega_s) &= -\tan(\delta) \cdot \tan(lat) \\ \omega_s &= \arccos(\omega_s) \end{aligned} \quad (3.34)$$

donde:

- $d_N$ : Día normal correspondiente 3.1.1
- lat: Latitud del emplazamiento

**3. Calcular los valores de  $T_m$  y  $T_r$ :** En el documento mencionado, el primer paso para llegar al perfil de temperaturas, una vez obtenidos los valores de  $T_{max}$  y  $T_{min}$  es calcular los dos valores intermedios:

$$T_m = \frac{T_{max} + T_{min}}{2} \quad (3.35)$$

$$T_r = \frac{T_{max} - T_{min}}{2} \quad (3.36)$$

**4. Calcular la matriz de  $T_a$ :** Una vez obtenidos los dos valores intermedios de  $T_m$  y  $T_r$ , el siguiente paso es calcular la matriz de valores de temperatura ambiental  $T_a$  para cada hora. Para ello el documento nos indica que debemos calcular tres valores que denominamos  $a_1$ ,  $a_2$ ,  $a_3$ . Utilizaremos estos tres valores para calcular  $T_1, T_2, T_3$ , eligiendo uno u otro en función del angulo solar. Las expresiones de estos tres valores se indican a continuación:

$$a_1 = \frac{12\pi \cdot (\omega_s - \omega)}{21\pi + 12 \cdot \omega_s} \quad (3.37)$$

$$a_2 = \frac{\pi \cdot (3\pi - 12\omega)}{3\pi - 12\omega} \quad (3.38)$$

$$a_3 = \frac{\pi \cdot (24\pi + 12 \cdot (\omega_s - \omega))}{21 \cdot \pi + 12 \cdot \omega_s} \quad (3.39)$$

donde  $\omega$  es el ángulo solar y  $\omega_s$  es el ángulo amanecer.

Con estos tres valores podemos calcular la otra terna de  $T_1, T_2$  y  $T_3$  con las siguientes ecuaciones:

$$T_1 = T_m - Tr \cdot \cos(a_1) \quad (3.40)$$

$$T_2 = T_m + Tr \cdot \cos(a_2) \quad (3.41)$$

$$T_3 = T_m - Tr \cdot \cos(a_3) \quad (3.42)$$

**5. Selección del valor clave:** en la matriz de temperatura ambiente introduciremos uno de los tres valores en función de si el ángulo solar es menor al del amanecer, si está entre en amanecer y  $\frac{\pi}{4}$  o si es menor que  $\frac{\pi}{4}$ .

En código, todo este proceso toma la siguiente forma:

```

1 const wp = Math.PI / 4;
2 const Tm = (Tmax + Tmin) / 2;
3 const Tr = (Tmax - Tmin) / 2;
4 const Ta = [];
5 const decl = 23.45 * Math.sin((2 * Math.PI * (normalDays[index] + 284)) / 365);
6 const cosWs = -Math.tan(deg2rad(decl)) * Math.tan(deg2rad(latitude));
7 const ws = -Math.acos(cosWs);
8 for (let h = -12; h < 12; h++) {
9   const w = Math.cos(deg2rad(h * 15));
10  const a1 = (Math.PI * 12 * (ws - w)) / (21 * Math.PI + 12 * ws);
11  const a2 = (Math.PI * (3 * Math.PI - 12 * w)) / (3 * Math.PI - 12 * ws);
12  const a3 = (Math.PI * (24 * Math.PI + 12 * (ws - w))) / (21 * Math.PI + 12 * ws);
13  const T1 = Tm - Tr * Math.cos(a1);
14  const T2 = Tm + Tr * Math.cos(a2);
15  const T3 = Tm - Tr * Math.cos(a3);
16  if (w <= ws) {
17    Ta.push(T1);
18  } else if (w > ws && w <= wp) {
19    Ta.push(T2);
20  } else if (w > wp) {
21    Ta.push(T3);
22  }
23}

```

### Extracto de código 3.16: Cálculo de la temperatura ambiente

Habiendo obtenido la matriz de temperatura ambiente, el siguiente paso es utilizar estos valores para calcular la temperatura de la célula  $T_c$  con la ecuación 3.32 necesaria para calcular la tensión de circuito abierto  $V_{oc}$ . En código sería:

```

1 const cellTempProfiles = tempProfiles.map(({ hourlyTa, Tmax, Tmin }, indexA) => {
2   const meanValue = radiationData.meanValues[indexA];
3   const TCPprofile = hourlyTa.map((temp, indexB) => {
4     const Gef = meanValue.hourlyValues[indexB].Gtilt * 1000;
5     const Tc = temp + (Gef * (moduleData.TONC - 20)) / 800;
6     return Tc; });
7   return { TCPprofile, month: index + 1, Tmax, Tmin }; });

```

---

Extracto de código 3.17: Cálculo de la temperatura de célula

Como se observa en el código, para cada uno de los valores horarios utilizamos la  $G_{ef}$  en el plano inclinado (en el código aparece como *Gtilt*) y la temperatura ambiente calculada anteriormente (en el código es *temp*) para calcular la  $T_c$ .

### Cálculo de $V_{oc}$ y $I_{sc}$

Conociendo ya los valores horarios de temperatura de la célula, procedemos a calcular la tensión de circuito abierto  $V_{oc}$  utilizando la expresión 3.43.

$$V_{oc}(T_c) = V_{oc}^* + (T_c - T_c^*) \cdot \frac{dV_{oc}}{dT_c} \cdot N_{cs} \quad (3.43)$$

En código quedaría de la siguiente forma:

```

1 const VocProfile = cellTempProfile.map(({ TCPProfile }, index) => {
2     const VocArray = TCPProfile.map((temp) => {
3         const Voc = moduleData.VocStar + (temp - moduleData.Tc)
4             * (-2.3 / 1000) * moduleData.Ncs;
5         return Voc;
6     });
7     return VocArray
8 });

```

Extracto de código 3.18: Cálculo de la tensión de circuito abierto

Por último, podemos calcular el valor de la corriente de cortocircuito  $I_{sc}$  que depende linealmente del valor de irradiancia según se indica en la expresión 3.44 que en código resulta de la siguiente forma:

$$I_{sc} = G_{ef} \cdot \frac{I_{sc}^*}{G^*} \quad (3.44)$$

```

1 const IscProfile = radiationData.meanValues.map(({ hourlyValues }) => {
2     const IscArray = hourlyValues.map(({ Gtilt }) => {
3         const IscValue = Gtilt * 1000 * (moduleData.IscStar / moduleData.Gstar);
4         return IscValue;
5     });
6     return IscArray;
7 });

```

Extracto de código 3.19: Cálculo de la corriente de cortocircuito

### Factor de forma variable

Una vez obtenidos los valores de  $V_{oc}$  y  $I_{sc}$ , el siguiente paso ha de ser calcular los valores de tensión y corriente en el punto de máxima potencia, pues es donde el generador estará entregando su máxima potencia, como su propio nombre indica, y por tanto es un punto de interés para el cálculo.

Existen dos metodologías de cálculo de dicho punto, uno de ellos significantemente más sencillo que el otro. Éste consiste en suponer que el Factor de Forma, definido en la expresión 3.29 es constante.

Si suponemos que FF es constante, se podrían extraer los valores de tensión y corriente en el punto de máxima potencia ya que si

$$FF = FF^* \quad (3.45)$$

entonces

$$\frac{I_{mpp} \cdot V_{vmpp}}{I_{sc} \cdot V_{oc}} = \frac{I_{mpp}^* \cdot V_{vmpp}^*}{I_{sc}^* \cdot V_{oc}^*} \quad (3.46)$$

pudiendo así obtener los valores de  $I_{mpp}$  y  $V_{vmpp}$ .

Sin embargo, hacer esta suposición nos aleja en cierta manera de ofrecer una estimación lo más próxima a la realidad posible.

Por tanto, para intentar aproximarnos lo máximo posible a una estimación acertada, eliminaremos dicha suposición y tendremos en cuenta la variación del factor de forma. Para ello, utilizaremos el siguiente proceso:

1. **Cálculo de la tensión térmica,  $V_t$ , a temperatura de la célula:** Calcularemos el valor de  $V_t$  a 25 °C con la expresión:

$$V_{tn} = \frac{V_t \cdot (273 + 25)}{300} \quad (3.47)$$

```
1 const Vtn = (moduleData.Vt * (273 + 25)) / 300;
```

Extracto de código 3.20: Cálculo de  $V_{tn}$

2. **Cálculo de  $R_s^*$ :** El segundo paso consiste en calcular el valor de resistencia en serie con los valores STC:

$$R_s^* = \frac{\frac{V_{oc}^*}{N_{cs}} - \frac{V_{mpp}^*}{N_{cs}} + m \cdot V_{tn} \cdot \ln \left( 1 - \frac{I_{sc}^*}{I_{mpp}^*} \right)}{\frac{I_{mpp}^*}{N_{cp}}} \quad (3.48)$$

En código quedaría:

```
1 const RsStar =
2   (moduleData.VocStar / moduleData.Ncs -
3    moduleData.VmppStar / moduleData.Ncs +
4    moduleData.m * Vtn * Math.log(1 - moduleData.ImppStar / moduleData.IscStar)) /
5    (moduleData.ImppStar / moduleData.Ncp);
```

Extracto de código 3.21: Cálculo de  $R_s^*$

3. **Cálculo de  $r_s$ :** Utilizando el valor  $R_s^*$  calculado en el paso anterior junto con los valores de  $V_{oc}$  y  $I_{sc}$  calculados anteriormente podemos calcular  $r_s$  que se utilizará más adelante en proceso. La ecuación que lo define viene dada por

$$r_s = R_s^* \cdot \left( \frac{N_{cs}}{N_{cp}} \cdot \frac{I_{sc}}{V_{oc}} \right) \quad (3.49)$$

El equivalente en código es

```

1 const rs = VocProfile.map((VocArray, arrIndex) => {
2     return VocArray.map((VocValue, valIndex) => {
3         return RsStar * ((moduleData.Ncs / moduleData.Ncp)
4             * (IscProfile[arrIndex][valIndex] / VocValue));
5     });
6 });

```

Extracto de código 3.22: Cálculo de  $r_s$

Como se puede observar en la línea 4 del código, el valor de  $I_{sc}$  viene acompañado por dos índices entre corchete, esto es porque en todo momento estamos trabajando con valores horarios de cada día promedio correspondiente a cada uno de los 12 meses del año, por tanto, estamos trabajando una matriz de 12x24, de ahí el uso de los dos índices.

4. **Cálculo de  $k_{oc}$ :** A continuación, utilizando los valores de temperatura ambiente obtenidos con anterioridad junto con la tensión de circuito abierto, calcularemos  $k_{oc}$  mediante la expresión:

$$k_{oc} = \frac{V_{oc}/N_{cs}}{m \cdot V_t \cdot \frac{T_c+273}{300}} \quad (3.50)$$

En código tendría este forma:

```

1 const koc = VocProfile.map((VocArray, arrIndex) => {
2     return VocArray.map((VocValue, valIndex) => {
3         return VocValue / moduleData.Ncs / (moduleData.m * ((moduleData.Vt *
4             (cellTempProfile[arrIndex].TCPProfile[valIndex] + 273)) / 300));
5     });
6 });

```

Extracto de código 3.23: Cálculo de  $k_{oc}$

Con éstos cálculos previos, éste método propone localizar el punto de máxima potencia de forma aproximada mediante las ecuaciones:

$$i_{mpp} = 1 - \frac{D_M}{k_{oc}} \quad (3.51)$$

$$v_{mpp} = 1 - \frac{\ln(k_{oc}/D_M)}{k_{oc}} - r_s \cdot i_{mpp} \quad (3.52)$$

donde:

$$D_M = D_{M0} + 2 \cdot r_s \cdot D_{M0}^2 \quad (3.53)$$

$$D_{M0} = \frac{k_{oc} - 1}{k_{oc} - \ln k_{oc}} \quad (3.54)$$

En código el proceso a seguir para llegar a los dos valores de  $i_{mpp}$  y  $v_{mpp}$  es el siguiente:

```

1 const DMO = koc.map((kocArray, arrIndex) => {
2   return kocArray.map((kocValue, valIndex) => {
3     return (kocValue - 1) / (kocValue - Math.log(kocValue));
4   });
5 });

```

Extracto de código 3.24: Cálculo de  $D_{M0}$

```

1 const DM = DMO.map((DMOArray, arrIndex) => {
2   return DMOArray.map((DMOValue, valIndex) => {
3     return DMOValue + 2 * rs[arrIndex][valIndex] * Math.pow(DMOValue, 2);
4   });
5 });

```

Extracto de código 3.25: Cálculo de  $D_M$

```

1 const impp = DM.map((DMArry, arrIndex) => {
2   return DMArry.map((DMValue, valIndex) => {
3     return 1 - DMValue / koc[arrIndex][valIndex];
4   });
5 });

```

Extracto de código 3.26: Cálculo de  $i_{mpp}$

```

1 const vmp = impp.map((imppArry, arrIndex) => {
2   return imppArry.map((imppValue, valIndex) => {
3     return 1 - Math.log(koc[arrIndex][valIndex] /
4       DM[arrIndex][valIndex]) / koc[arrIndex][valIndex] -
5       rs[arrIndex][valIndex] * imppValue;
6   });
7 });

```

Extracto de código 3.27: Cálculo de  $v_{mpp}$

Por último, multiplicando los valores de  $i_{mpp}$  y  $v_{mpp}$  por  $I_{sc}$  y  $V_{oc}$  respectivamente, obtendremos los valores de  $I_{mpp}$  y  $V_{mpp}$  que serán los que utilizaremos para calcular la potencia entregada por el generador en el punto de máxima potencia.

```

1 const Vmpp = vmpp.map((vmppArray, arrIndex) => {
2   return vmppArray.map((vmppValue, valIndex) => {
3     return vmppValue * VocProfile[arrIndex][valIndex];
4   });
5 });
6 const Impp = impp.map((imppArray, arrIndex) => {
7   return imppArray.map((imppValue, valIndex) => {
8     return imppValue * IscProfile[arrIndex][valIndex];
9   });
10 });
11

```

Extracto de código 3.28: Cálculo de  $V_{mpp}$  y  $I_{mpp}$

Una vez calculados los valores de  $I_{mpp}$  y  $V_{mpp}$  podemos pasar a obtener los valores horarios de la potencia en el MPP, simplemente multiplicando los valores de corriente y tensión:  $P_{mpp} = I_{mpp} \cdot V_{mpp}$ .

```

1 const calculatePmpp = (ImppProfile, VmppProfile) => {
2   const PmppProfile = ImppProfile.map((ImppArray, dayIndex) => {
3     const PmppArray = ImppArray.map((ImppValue, hourIndex) => {
4       return ImppValue * VmppProfile[dayIndex][hourIndex];
5     });
6     return PmppArray;
7   });
8   return PmppProfile;
9 };
10

```

Extracto de código 3.29: Cálculo de  $P_{mpp}$

### 3.3.5. Cálculo final de potencias y energías

La potencia que obtenemos es la de un solo módulo. Para conocer la potencia que va a ser capaz de entregar el generador, debemos tener en cuenta su configuración de módulos en serie y en paralelo, y multiplicar estos valores de la siguiente manera:

```

1 const calculateGeneratorPower = (PmppProfile) => {
2   const PdcProfile = PmppProfile.map((PmppArray) => {
3     return PmppArray.map((PmppValue) => {
4       return PmppValue * generatorData.seriesModules * generatorData.parallelModules;
5     });
6   });
7   return PdcProfile;
8 };
9

```

Extracto de código 3.30: Cálculo de la potencia del generador

Con este paso obtenemos la potencia horaria entregada por el generador fotovoltaico. El siguiente paso será pasar esa potencia a través del inversor y calcular la potencia a la salida de este. Para ello debemos llevar a cabo el siguiente desarrollo.

Lo primero, establecemos las expresiones de las potencias normalizadas Siendo  $P_{inv}$  la potencia nominal del inversor:

$$p_i = \frac{P_{DC}}{P_{inv}} \quad (3.55)$$

$$p_o = \frac{P_{AC}}{P_{inv}} \quad (3.56)$$

Para calcular la estos dos valores, utilizaremos un modelo empleado en inversores fotovoltaicos:

$$\eta_{inv} = \frac{p_o}{p_o + k_0 + k_1 p_o + k_2 p_o^2} \quad (3.57)$$

donde:  $k_0$ ,  $k_1$  y  $k_2$  son tres coeficientes ofrecidos por el fabricante del inversor.

Por otro lado, por definición de rendimiento, tenemos:

$$\eta_{inv} = \frac{p_o}{p_i} \quad (3.58)$$

Si despejamos las dos ecuaciones anteriores:

$$p_i = p_o + k_0 + k_1 p_o + k_2 p_o^2 \quad (3.59)$$

A partir de la expresión anterior obtenemos una ecuación de segundo grado con  $p_o$  como incógnita:

$$k_2 p_o^2 + (k_1 + 1)p_o + (k_0 - p_i) = 0 \quad (3.60)$$

Y por último, si conocemos la  $p_o$  podemos calcular la potencia en corriente alterna que entrega el generador:

$$P_{AC} = p_o \cdot p_{inv} \quad (3.61)$$

En código, el desarrollo quedaría de la siguiente manera:

1. **Cálculo de  $p_i$ :** A partir de la potencia calculada anteriormente y conociendo la potencia del inversor, podemos calcular la  $p_i$  de esta manera:

```

1 const calculatePiProfile = (PdcProfile) => {
2   return PdcProfile.map((PdcArray) => {
3     return PdcArray.map((PdcValue) => {
4       return PdcValue / inverterData.power;
5     });
6   });
7 };

```

Extracto de código 3.31: Cálculo de  $p_i$

2. **Cálculo de  $p_o$ :** Utilizando  $p_i$  podemos despejar la ecuación de segundo grado y calcular el valor de  $p_o$  eligiendo la solución que cumpla que  $p_o > 0$ :

```

1 const calculatePoProfile = (PiProfile) => {
2   return PiProfile.map((PiArray) => {
3     return PiArray.map((PiValue) => {
4       if (PiValue <= 0) {
5         return 0;
6       }
7       const a = inverterData.k2;
8       const b = inverterData.k1 + 1;
9       const c = inverterData.k0 - PiValue;
10      const s1 = (-b + Math.sqrt(b * b - 4 * a * c)) / (2 * a);
11      const s2 = (-b - Math.sqrt(b * b - 4 * a * c)) / (2 * a);
12      return s1 > 0 ? s1 : s2;
13    });
14  });
15};

```

Extracto de código 3.32: Cálculo de  $p_o$

3. **Cálculo de  $P_{inv}$ :** Habiendo calculado la  $p_o$ , podemos despejar la potencia generada por el generador ( $P_{inv}$ ) de la ecuación 3.56:

```

1 const calculatePacProfile = (PoProfile) => {
2   return PoProfile.map((PoArray) => {
3     return PoArray.map((PoValue) => {
4       return PoValue * inverterData.power;
5     });
6   });
7 };

```

Extracto de código 3.33: Cálculo de  $P_{AC}$

Finalmente, conociendo las potencias en corriente continua y en alterna, podemos calcular la energía mensual y anual que es capaz de producir el generador configurado. Para ello, en el caso del balance mensual, sumaremos los 24 valores horario que hemos obtenido, y los multiplicaremos por 30.

Por otro lado, para calcular la energía anual, simplemente sumaremos los 12 valores de energía mensual que hemos calculado.

```

1 const calculateMonthlyEac = (PacProfile) => {
2   return PacProfile.map((PacArray) => {
3     return (
4       30 * PacArray.reduce((total, currentValue) => {
5         return total + (currentValue > 0 ? currentValue : 0);
6         EacCell.textContent = MonthlyEac[index];
7       }, 0)))};

```

Extracto de código 3.34: Cálculo de la energía mensual en alterna

```

1 const calculateMonthlyEdc = (PdcProfile) => {
2   return PdcProfile.map((PdcArray) => {
3     return (
4       30 * PdcArray.reduce((total, currentValue) => {
5         return total + (currentValue > 0 ? currentValue : 0);
6       }, 0)
7     );
8   });
9 };

```

Extracto de código 3.35: Cálculo de la energía mensual en continua

```
1 const calculateAnualEnergy = (MonthlyEnergy) => {  
2     return MonthlyEnergy.reduce((total, currentValue) => {  
3         return total + (currentValue > 0 ? currentValue : 0);  
4     }, 0); };
```

Extracto de código 3.36: Cálculo de la energía anual

# Capítulo 4

## Ejemplo práctico de aplicación

Una vez llevado a cabo el desarrollo teórico de la aplicación y los cálculos necesarios para llegar a los resultados deseados, el siguiente paso será aplicar estos a un emplazamiento concreto, observar los resultados obtenidos y analizarlos para sacar una conclusión.

Éste capítulo también servirá como una guía detallada de uso de la aplicación, explicando la información a introducir en cada paso y como obtenerlos.

La aplicación está disponible a través del enlace <http://solarcalc.app> para utilizarse de manera totalmente gratuita.

### 4.1. Obtención de datos del usuario

El formulario que se le presenta al usuario para que éste introduzca la información de su emplazamiento es el siguiente:

**Datos del emplazamiento**

**Dirección**

**Municipio**      **Código postal**

**Datos de la casa**

**Inclinación del tejado en grados**      **Área del tejado en metros cuadrados**

**Orientación**      **Nivel de suciedad**

**Enviar**

Figura 4.1: Formulario web para obtención de datos

Como se puede observar, el formulario consta de dos partes, por un lado está la dirección como tal,

que se utilizará para obtener las coordenadas del emplazamiento y por otro está la configuración física de la superficie destinada a la instalación de los paneles solares.

#### 4.1.1. Latitud y longitud a partir de la dirección

La primera parte del formulario consiste en obtener la información acerca del emplazamiento del usuario, es decir, su latitud y longitud. Estos datos son los que se utilizarán en el código, para obtener la información de la radiación en dicho lugar. El proceso completo para obtener los datos de radiación se describe en 5.2.3

**Datos del emplazamiento**

<b>Direccion</b>	
<input type="text"/> Calle	
<b>Municipio</b>	<b>Código postal</b>
<input type="text"/> Municipio	<input type="text"/> # Código postal

Figura 4.2: Campos del formulario para la dirección

Para ello, el formulario pide una dirección, que no necesariamente tiene que ser la exacta del emplazamiento, sino que se puede utilizar una dirección genérica cercana al lugar donde se va a realizar la instalación, ya que para una distancia relativamente corta la radiación solar no variará de manera notable.

En la sección 5.2.1 se expone en detalle el proceso para obtener las coordenadas en función de la dirección del usuario.

En este caso práctico la dirección que utilizaremos será la de Guadalajara. Para ésta dirección, las coordenadas que Google nos devuelve son:

- **Latitud:**  $40,632^\circ$ .
- **Longitud:**  $-3,166^\circ$ .

#### 4.1.2. Datos de la superficie destinada a la instalación

Una vez obtenidas las coordenadas del emplazamiento, es necesario conocer la inclinación, la orientación y el nivel de suciedad de la superficie, para poder realizar el cambio de la radiación en el plano horizontal obtenida anteriormente a la radiación eficiente incidente.

Datos de la superficie

Inclinación del tejado en grados	Área del tejado en metros cuadrados
<input type="text"/> Inclinación del tejado	<input type="text"/> Área del tejado
Orientación	Nivel de suciedad
<input type="text"/> 0	<input type="text"/> Nivel bajo

Figura 4.3: Campos del formulario para los datos de la superficie

El campo del nivel de suciedad es un desplegable ya que solamente existen 4 niveles, que afectan a los cálculos de una manera específica, como se ha visto en el capítulo teórico, por tanto, debemos limitar la forma en la que el usuario introduce esa información.

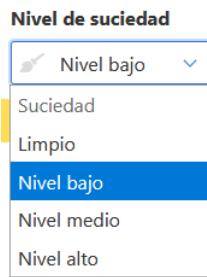


Figura 4.4: Campos del formulario para los datos de la superficie

Para ello el formulario nos ofrece 4 campos de información:

- **Inclinación en grados:** Para obtener la inclinación de la superficie en grados se puede utilizar cualquier smartphone actual dotado con un sensor giroscópico y una aplicación de tipo nivel, algún dispositivo de medición de ángulos como un transportador.
- **Orientación:** Para la obtener la orientación de la superficie se puede utilizar una brújula, ya sea analógica o digital. Se considera que 0 grados es el SUR.
- **Área del tejado:** Una estimación aproximada del área disponible el área que se desea cubrir con paneles solares.
- **Nivel de suciedad:** Este campo es el mas subjetivo y será a criterio del usuario. Se recomienda realizar cálculos con el peor caso para establecer un margen pero, no obstante, si se conoce con seguridad cuál es el nivel de suciedad de la zona, se ha de usar éste.

Un proceso detallado de como obtener los valores de inclinación y orientación se describe a continuación

#### 4.1.3. Obtención de la inclinación y orientación de la superficie

##### Medición de la inclinación

En la aplicación se presentan dos campos dentro del formulario que debe llenar el usuario, que requieren introducir la inclinación y la orientación de la superficie donde se va a instalar el

generador fotovoltaico.

Para ello, existen multitud de aplicaciones gratuitas en la tienda de aplicaciones tanto de Android como de iOS. En este caso, se va a mostrar un ejemplo con un dispositivo Android, pero esto se puede trasladar igualmente a un dispositivo iOS.

La aplicación que se va a utilizar se llama *Angle Meter*, y la empresa que lo desarrolla se llama *Smart Tool Factory*. Se encuentra en la Play Store de Android de manera gratuita.

Cuando se abre la aplicación por primera vez, encontramos lo siguiente:

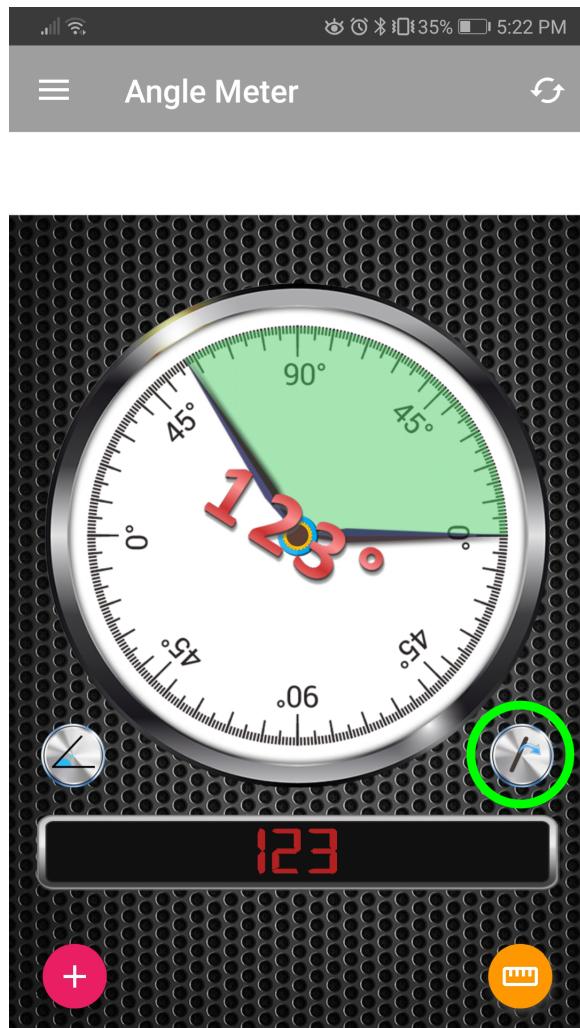


Figura 4.5: Pantalla de inicio de la aplicación

A continuación, pulsamos el botón que sale remarcado con un círculo verde, lo que hará que cambie el modo de medición de ángulos a inclinación, que es el necesario para obtener el dato de la inclinación. Cuando la aplicación cambie a ese modo, se verá de la siguiente manera:

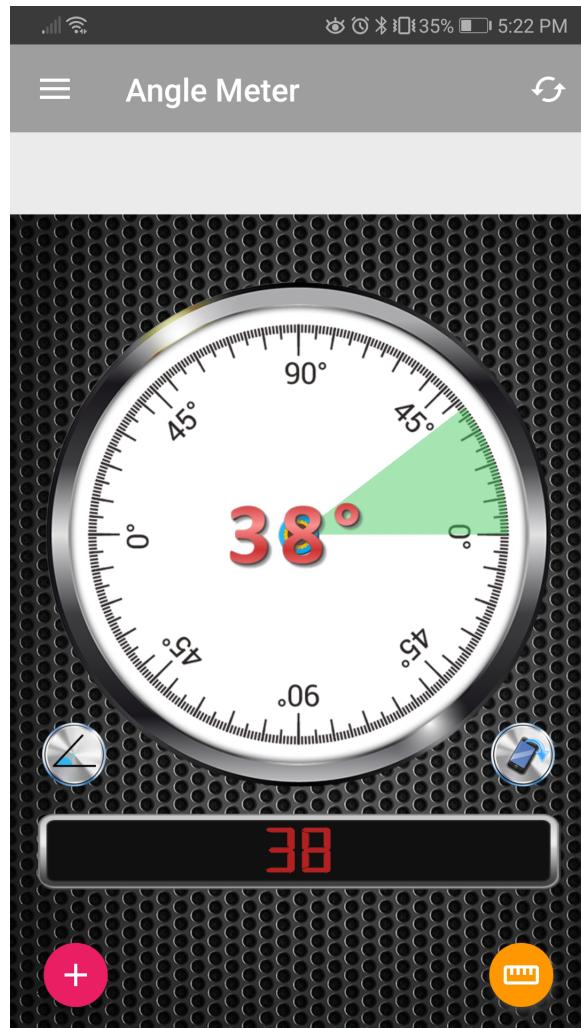


Figura 4.6: Medición de ángulo

A continuación, el dispositivo se debe colocar según la figura que se muestra abajo, teniendo en cuenta que la imagen se muestra con una perspectiva perpendicular a la superficie.

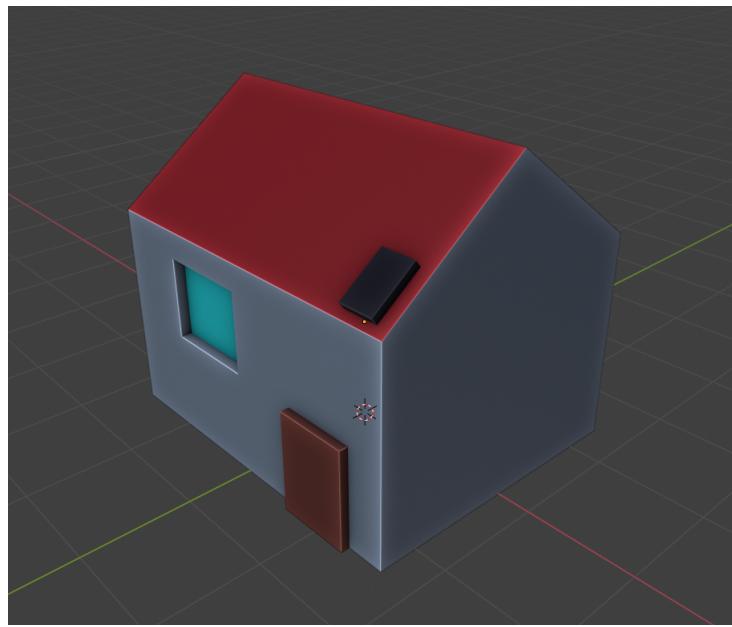


Figura 4.7: Posición del dispositivo

De esta manera, obtenemos la inclinación de la superficie en grados.

### Medición de la orientación

Para obtener la orientación de la superficie, debemos cambiar una vez más el modo de funcionamiento de la aplicación, dejando el dispositivo en la misma posición que en la medición de inclinación.

Para ello, abrimos el menú desde la izquierda y pulsamos en la opción llamada *Compass*, como se muestra en la siguiente figura.

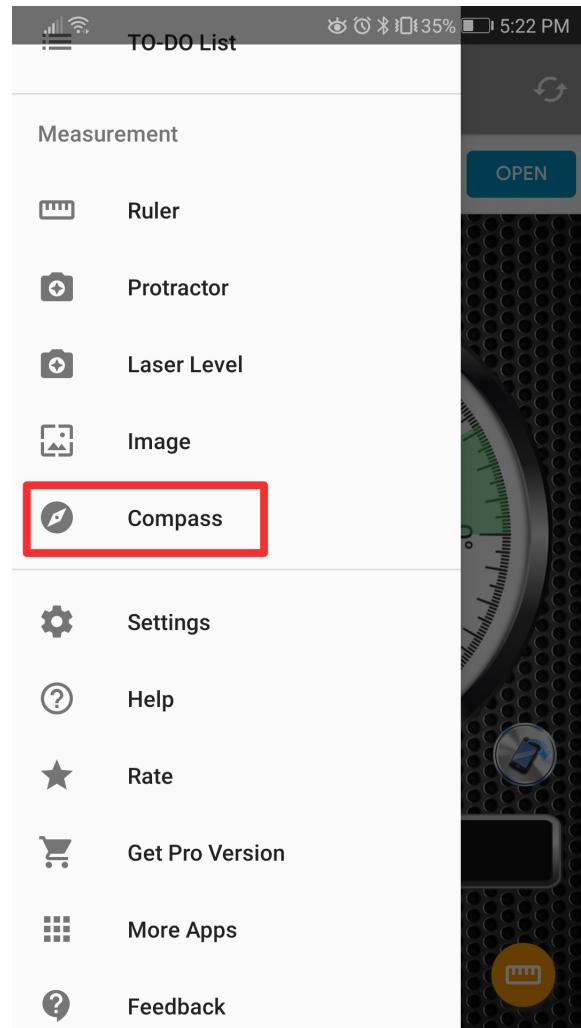


Figura 4.8: Opción brújula del menú

Una vez abierta la brújula, observamos la medición que nos muestra en la parte inferior central de la pantalla. A esa medición debemos restarle  $180^{\circ}$  para obtener el valor que necesita la aplicación de cálculo. Es decir, si observamos una orientación de  $193^{\circ}$ , en el formulario introduciremos  $13^{\circ}$ . En la figura de abajo se muestra como se ve la pantalla de la brújula.

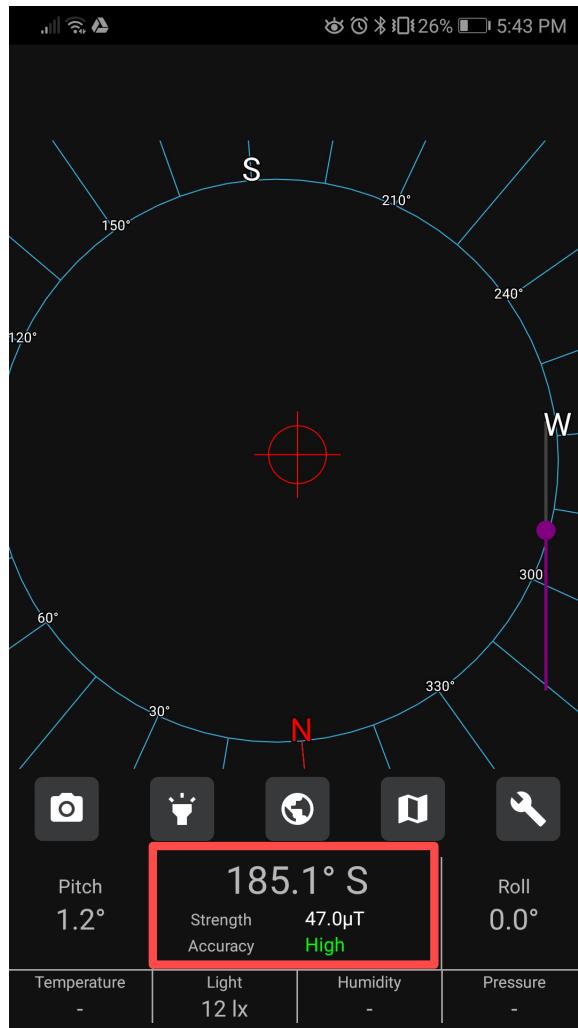


Figura 4.9: Opción brújula del menú

## 4.2. Aplicación de los datos al proceso de cálculo

En la sección anterior hemos recogido los datos que un usuario ha introducido a través del formulario. Estos datos son:

- **Latitud:**  $40,632^\circ$ .
- **Longitud:**  $-3,166^\circ$ .
- **Inclinación de la superficie:**  $20^\circ$ .
- **Área de la superficie:**  $40m^2$ .
- **Orientación de la superficie:**  $30^\circ$  SUR.
- **Nivel de suciedad:** Medio.

A continuación, utilizaremos estos datos para recorrer numéricamente el proceso teórico descrito en el capítulo anterior.

#### 4.2.1. Valores medios mensuales de radiación global

Tal y como se menciona en la sección 3.1.2, el primer paso para poder llevar a cabo el proceso de cálculo de la radiación incidente efectiva es obtener la radiación global media para cada uno de los doce meses, en el emplazamiento indicado por el usuario.

Para las coordenadas introducidas por el usuario, el punto con información mas cercano que tenemos es de latitud  $40,57^{\circ}$  y longitud  $-3,16^{\circ}$ .

El proceso de obtención de los siguientes valores se describe en detalle en la sección 5.2.3.

Para este punto, los valores de irradiación media son:

$kWh/m^2$	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic
Valor medio	2,0	3,1	4,8	5,7	6,8	8,0	7,8	6,8	5,1	3,5	2,2	1,7

Cuadro 4.1: Irradiación global media mensual



Figura 4.10: Radiación global en el emplazamiento

#### 4.2.2. Irradiancia extra-terrestre diaria

Además, por otro lado, para poder continuar con el proceso de cálculo, es necesario calcular la irradiancia extra-terrestre diaria, utilizando la latitud indicada por el usuario y los días promedio de la tabla 3.1.1. En las ecuaciones de la sección 3.1.1 se indican los pasos a seguir y el resultado es:

$kW/m^2$	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic
$B_{0d}(0)$	4,12	5,48	7,47	9,57	11,01	11,59	11,26	10,01	8,05	5,90	4,30	3,58

Cuadro 4.2: Irradiancia extra-terrestre diaria



Figura 4.11: Irradiación extra-terrestre

#### 4.2.3. Separación de la radiación global horizontal en sus componentes

Una vez conocidos los valores mensuales de la radiación global en el plano horizontal, el siguiente paso se centra en separar la radiación global en sus dos componentes, la directa y la difusa, tal y como se explica en la sección 3.1.2.

Como podemos observar en la ecuación 3.4, el primer paso para separar la radiación global en sus dos componentes es calcular el índice de claridad, para que posteriormente calculemos la fracción de radiación difusa.

Si aplicamos dicha ecuación a los resultados anteriores obtenemos los siguientes resultados:

	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic
$K_{Td}$	0,485	0,565	0,642	0,595	0,617	0,690	0,692	0,679	0,632	0,592	0,511	0,461

Cuadro 4.3: Índice de claridad

A continuación, utilizando la ecuación de Page 3.6, con el índice de claridad calculado, podemos calcular la fracción de difusa:

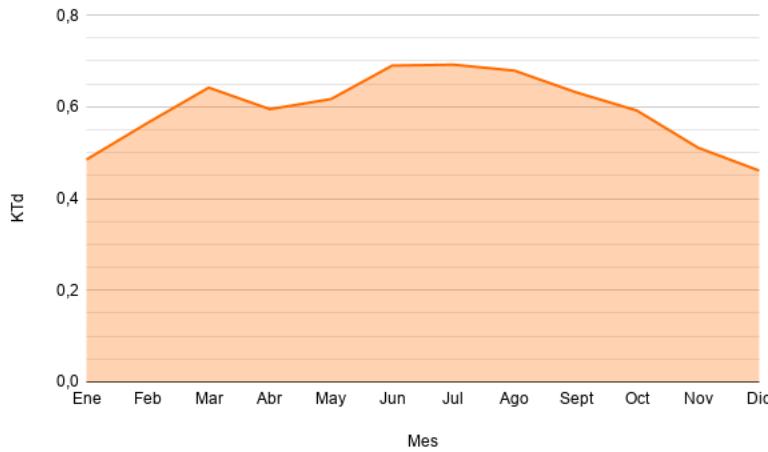


Figura 4.12: Índice de claridad

	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic
$F_D$	0,452	0,361	0,274	0,327	0,302	0,220	0,218	0,232	0,285	0,330	0,421	0,478

Cuadro 4.4: Fracción de difusa



Figura 4.13: Fracción de difusa

Como era de esperar, se observa que la fracción de difusa es más elevada en los meses con mayor nubosidad y menos intensidad solar.

### Cálculo de las componentes de la radiación en el plano horizontal

Con estos valores, utilizando las ecuaciones 3.7 y 3.8 podemos obtener las dos componentes de la radiación en el plano horizontal.

$kWh/m^2$	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic
$B_d(0)$	1,096	1,980	3,484	3,835	4,745	6,239	6,103	5,220	3,647	2,344	1,272	0,886
$D_d(0)$	0,904	1,120	1,316	1,865	2,055	1,761	1,697	1,580	1,453	1,156	0,928	0,814

Cuadro 4.5: Irradiación directa y difusa en el plano horizontal

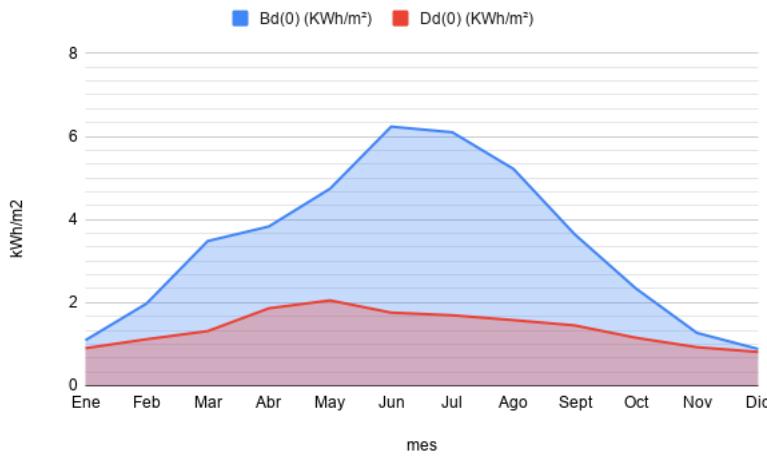


Figura 4.14: Componentes de la irradiación en el plano horizontal

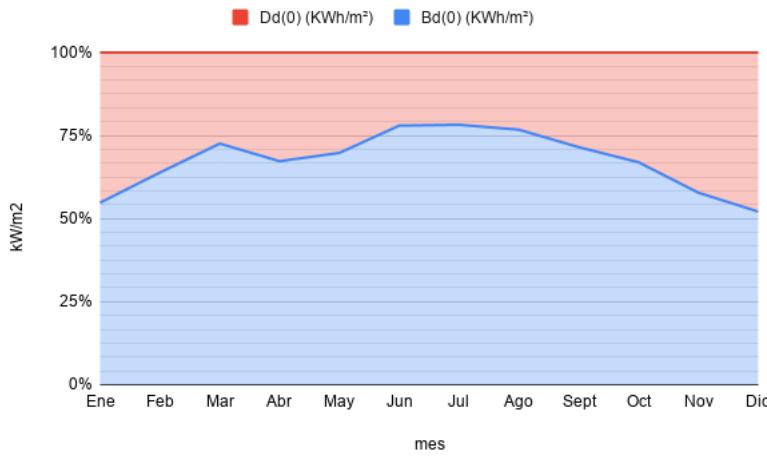


Figura 4.15: Componentes de la irradiación en el plano horizontal

La primera gráfica muestra los valores absolutos de las dos componentes, que la segunda muestra el porcentaje de cada una de las dos componentes. Una vez más, se observa una aportación mayor de la radiación difusa en los meses de invierno.

#### 4.2.4. Irradiancia en la superficie inclinada

El término irradiancia ( $W/m^2$ ) se refiere a la magnitud utilizada para describir la potencia incidente por unidad de superficie de cualquier tipo de radiación, en este caso la solar. En cambio, el término irradiación ( $Wh/m^2$ ) es una energía por unidad de área.

Sin embargo, como la evolución de la radiación solar a lo largo de una hora es relativamente lenta, podemos asumir que la irradiación solar durante esa hora coincide con el valor media de la irradiancia de dicha hora y utilizar los dos términos de manera equivalente.

#### Irradiancia en el plano horizontal

Conociendo ya los valores diarios de la radiación directa y difusa en el plano horizontal, el próximo paso es obtener los valores horarios de irradiancia, tanto directa como difusa, en el plano horizontal para poder llevar a cabo posteriormente, la traslación de estos valores al plano inclinado. El proceso se describe detalladamente en las secciones 3.2.1 y 3.2.2.

Al tratarse de 24 valores horarios, para cada uno de los 12 meses, en total tendríamos 288 valores. Por tanto, y con el fin de no saturar este documento con demasiado valores, se van a indicar los resultados de solamente uno de los meses, en concreto de julio. Estos resultados se muestran en la tabla 4.6

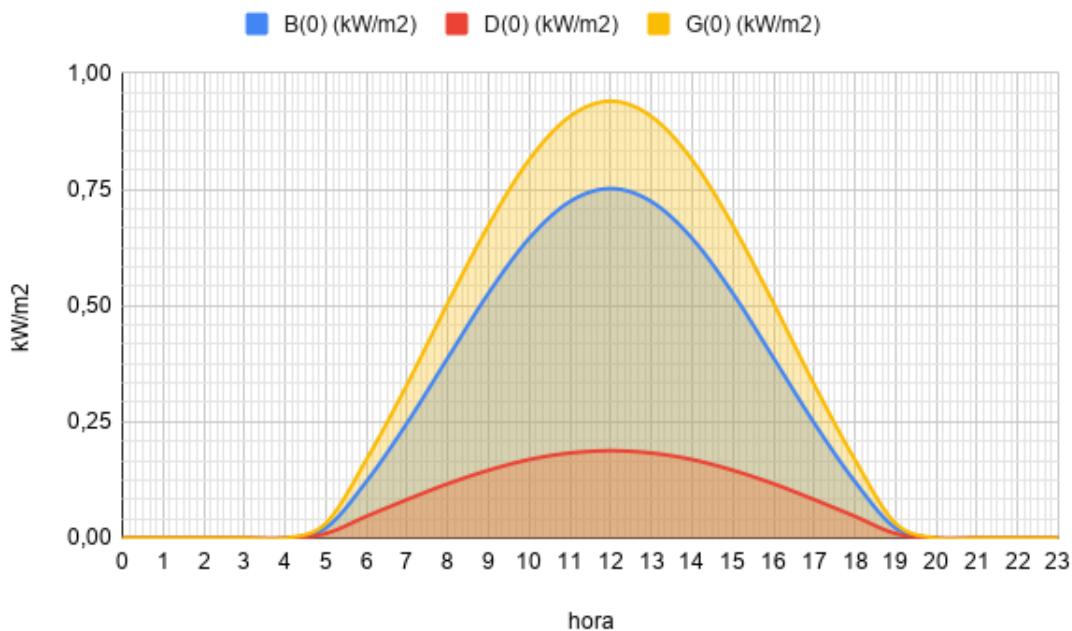


Figura 4.16: Irradiancia para el día promedio del mes de julio

hora	$r_D$	$r_G$	$B(0)$ ( $kW/m^2$ )	$D(0)$ ( $kW/m^2$ )	$G(0)$ ( $kW/m^2$ )
0	-0,056	-0,027	0	0	0
1	-0,053	-0,026	0	0	0
2	-0,045	-0,024	0	0	0
3	-0,031	-0,018	0	0	0
4	-0,014	-0,009	0	0	0
5	0,006	0,004	0,022	0,010	0,032
6	0,027	0,022	0,122	0,047	0,169
7	0,049	0,042	0,248	0,083	0,331
8	0,069	0,065	0,388	0,117	0,506
9	0,086	0,086	0,527	0,146	0,673
10	0,099	0,104	0,645	0,169	0,814
11	0,108	0,116	0,724	0,183	0,907
12	0,111	0,121	0,752	0,188	0,940
13	0,108	0,116	0,724	0,183	0,907
14	0,099	0,104	0,645	0,169	0,814
15	0,086	0,086	0,527	0,146	0,673
16	0,069	0,065	0,388	0,117	0,506
17	0,049	0,042	0,248	0,083	0,331
18	0,027	0,022	0,122	0,047	0,169
19	0,006	0,004	0,022	0,010	0,032
20	-0,014	-0,009	0	0	0
21	-0,031	-0,018	0	0	0
23	-0,045	-0,024	0	0	0
23	-0,053	-0,026	0	0	0

Cuadro 4.6: Irradiancia directa, difusa y global en el plano horizontal para el día promedio del mes de julio

### Traslación de los valores al plano inclinado

Una vez obtenidos los valores horarios en el plano horizontal, podemos trasladar estos al plano inclinado con el proceso descrito de la ecuación 3.17 hasta la ecuación 3.23.

hora	$B(\beta, \alpha)$ (kW/m <sup>2</sup> )	$D^C(\beta, \alpha)$ (kW/m <sup>2</sup> )	$D^I(\beta, \alpha)$ (kW/m <sup>2</sup> )	$D(\beta, \alpha)$ (kW/m <sup>2</sup> )	$G(\beta, \alpha)$ (kW/m <sup>2</sup> )
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0,006	0,006	0,006
6	0	0	0,027	0,027	0,027
7	0,12	0,018	0,044	0,062	0,184
8	0,280	0,042	0,057	0,099	0,379
9	0,448	0,067	0,065	0,132	0,580
10	0,603	0,091	0,069	0,160	0,763
11	0,721	0,108	0,072	0,180	0,901
12	0,786	0,118	0,072	0,190	0,977
13	0,787	0,119	0,072	0,191	0,978
14	0,724	0,109	0,070	0,179	0,903
15	0,610	0,092	0,065	0,156	0,767
16	0,463	0,069	0,057	0,127	0,589
17	0,304	0,046	0,044	0,090	0,394
18	0,156	0,023	0,027	0,050	0,206
19	0,033	0,005	0,006	0,011	0,044
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0

Cuadro 4.7: Irradiancia directa, difusa y global en el plano inclinado, para el día promedio del mes de julio

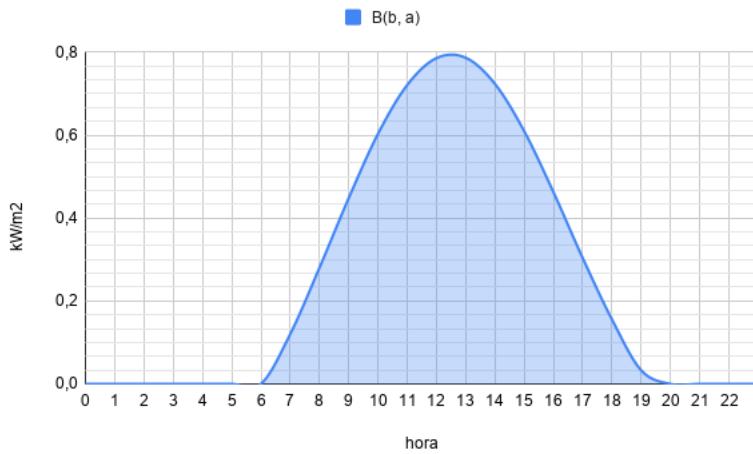


Figura 4.17: Irradiancia directa en el plano inclinado

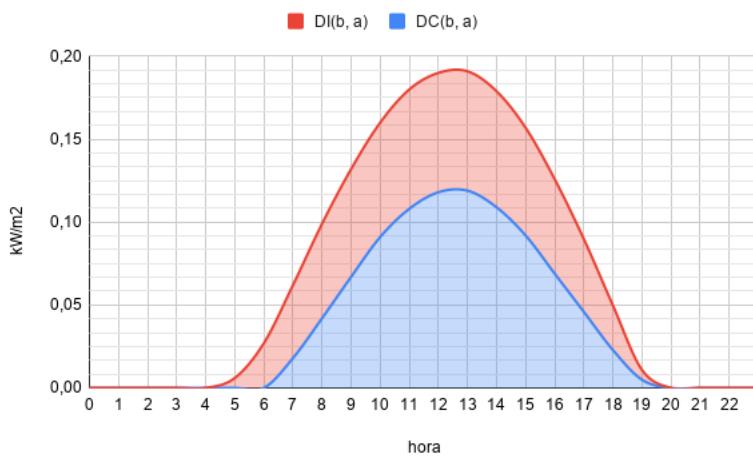


Figura 4.18: Componentes de la irradiancia difusa en el plano inclinado

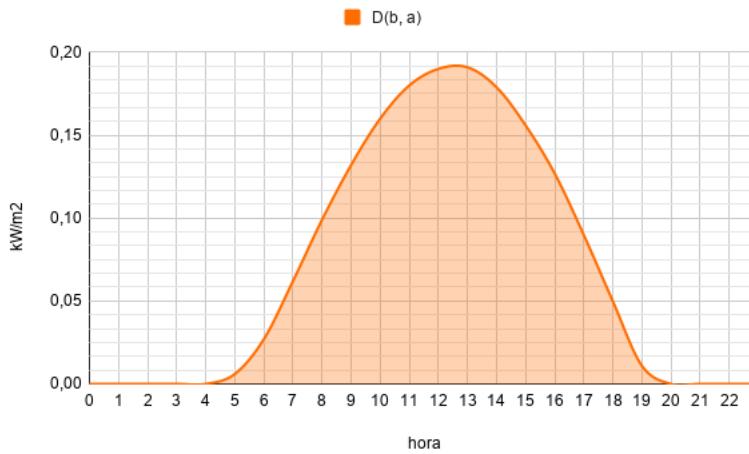


Figura 4.19: Irradiancia difusa en el plano inclinado

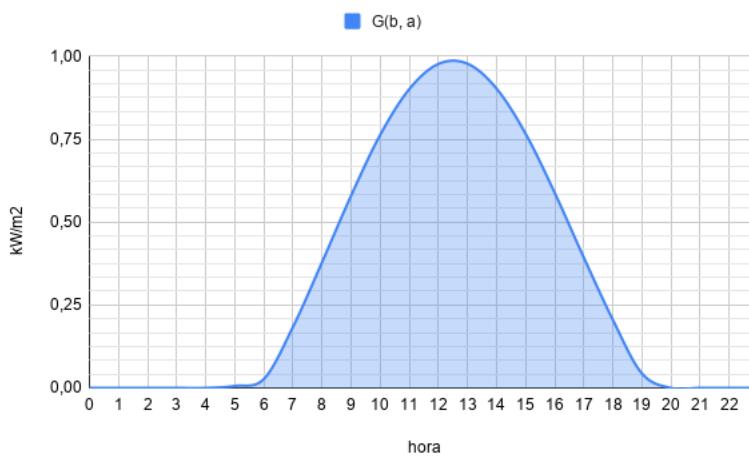


Figura 4.20: Irradiancia global en el plano inclinado

Como se puede observar, a diferencia de la irradiancia en el plano horizontal, estos resultados no son simétricos respecto al mediodía ya que ahora está afectando la orientación de la superficie, que no está orientada totalmente hacia el sur.

#### 4.2.5. Pérdidas por ángulo de incidencia y suciedad

Habiendo calculado los valores de irradiancia horaria en la superficie inclinada, el siguiente paso consiste en aplicar las pérdidas por ángulo de incidencia y suciedad, mediante el proceso descrito en la sección 3.2.3. Los resultados del proceso se muestran en la tabla 4.8.

hora	$B_{ef}(\beta, \alpha)$ (kW/m <sup>2</sup> )	$D_{ef}^C(\beta, \alpha)$ (kW/m <sup>2</sup> )	$D_{ef}^I(\beta, \alpha)$ (kW/m <sup>2</sup> )	$D_{ef}(\beta, \alpha)$ (kW/m <sup>2</sup> )	$G_{ef}(\beta, \alpha)$ (kW/m <sup>2</sup> )
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0
5	0	0	0,006	0,006	0,006
6	0	0	0,025	0,025	0,025
7	0,076	0,011	0,041	0,052	0,128
8	0,243	0,037	0,052	0,089	0,3314
9	0,422	0,064	0,059	0,123	0,545
10	0,583	0,088	0,064	0,152	0,735
11	0,704	0,106	0,066	0,172	0,8767
12	0,770	0,116	0,066	0,182	0,952
13	0,771	0,116	0,066	0,182	0,953
14	0,708	0,106	0,064	0,170	0,879
15	0,593	0,089	0,059	0,149	0,742
16	0,443	0,067	0,052	0,119	0,562
17	0,277	0,042	0,041	0,082	0,349
18	0,119	0,018	0,025	0,043	0,162
19	0,010	0,001	0,006	0,007	0,017
20	0	0	0	0	0
21	0	0	0	0	0
22	0	0	0	0	0
23	0	0	0	0	0

Cuadro 4.8: Irradiancia directa, difusa y global efectiva incidente en el plano inclinado, para el día promedio del mes de julio

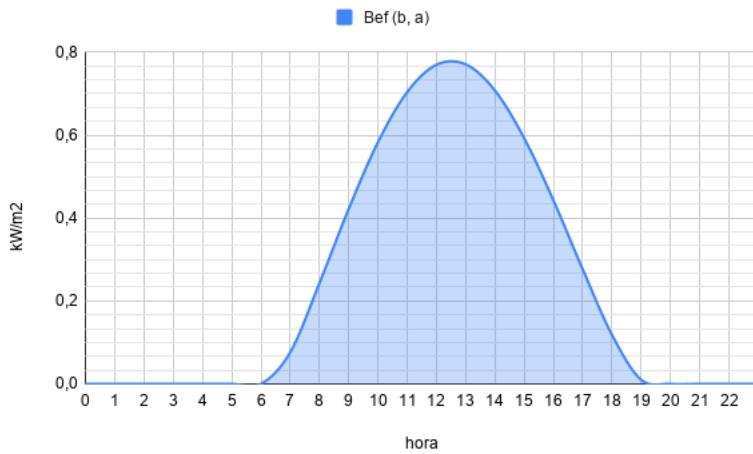


Figura 4.21: Irradiancia directa efectiva en el plano inclinado

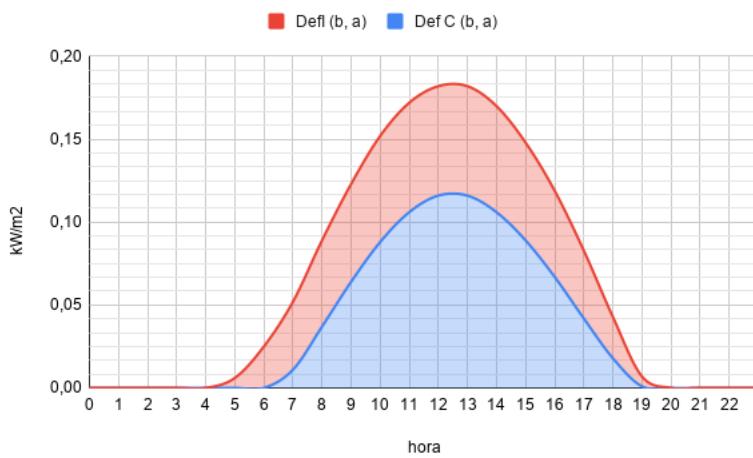


Figura 4.22: Componentes de la irradiancia difusa efectiva en el plano inclinado

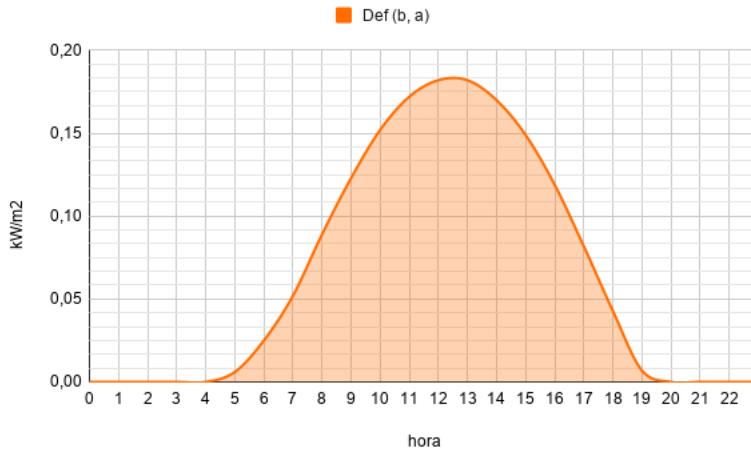


Figura 4.23: Irradiancia difusa efectiva en el plano inclinado

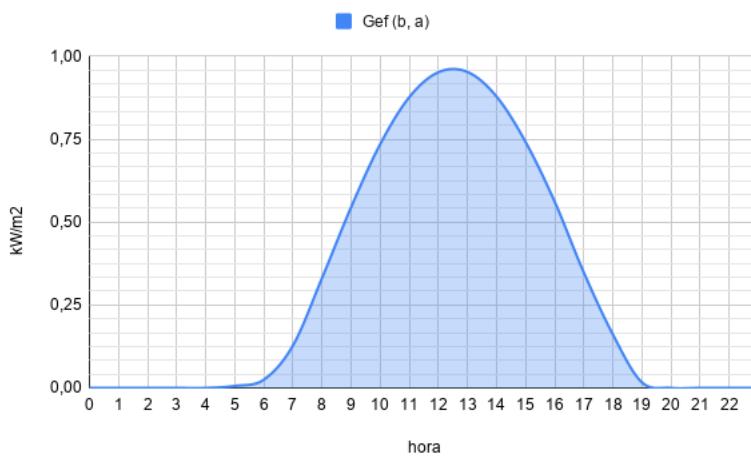


Figura 4.24: Irradiancia global efectiva en el plano inclinado

Como se puede apreciar, los valores eficaces son ligeramente inferiores al aplicar las pérdidas por el nivel de suciedad medio y el ángulo de incidencia.

#### 4.2.6. Configuración del módulo solar y su comportamiento

La primera parte del proceso de cálculo está completada, teniendo ya los valores eficaces de irradiancia horaria para cada uno de los 12 días promedio.

La segunda parte consiste en aplicar dicha radiación a un módulo para estimar la potencia máxima que será capaz de entregar, así como la energía que producirá al cabo de un año.

Para ello, tal y como se describe en el desarrollo teórico, debemos calcular la potencia en el punto de máxima potencia con las condiciones de temperatura y radiación del emplazamiento. Como

ya tenemos los datos de radiación, a continuación calcularemos el perfil horario de temperatura como se describe en la sección de la página 26.

El resultado de ese proceso, para las coordenadas indicadas por el usuario se muestran en las tablas 4.9 y 4.10.

$^{\circ}\text{C}$	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic
$T_{\min}$	0,7	2,6	5,8	8,9	11,8	17,5	22,4	18,6	14,1	10,8	6,8	4,9
$T_{\max}$	11,4	10,6	17,1	20,3	27	31,1	36,4	32,3	26,4	20,3	17,5	14,2

Cuadro 4.9: Temperaturas máximas y mínimas en  $^{\circ}\text{C}$  para cada uno de los 12 meses

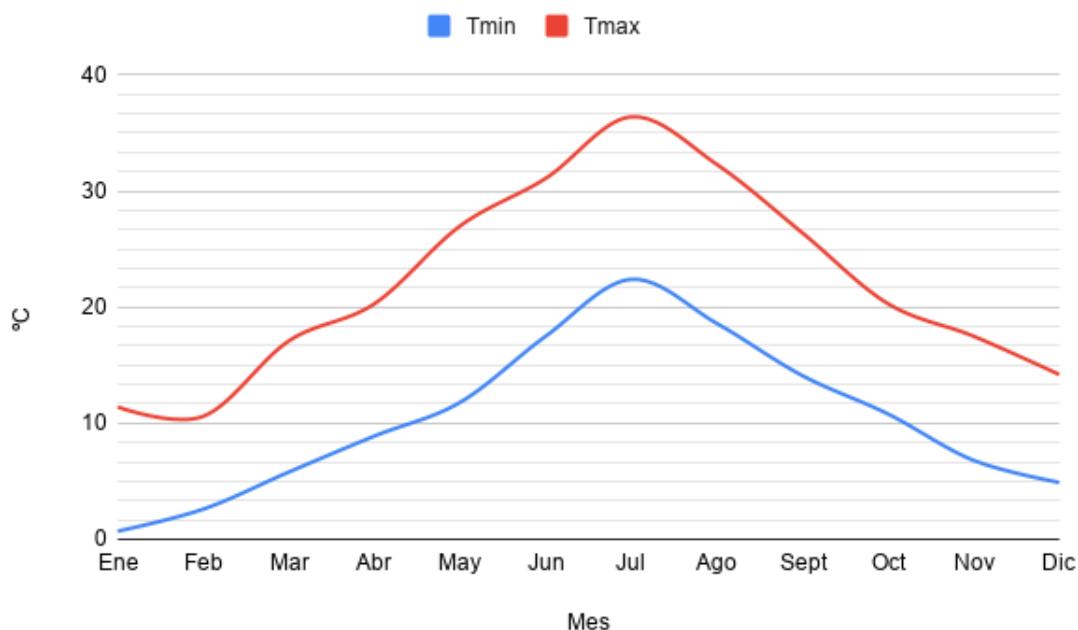


Figura 4.25: Temperaturas máximas y mínimas en  $^{\circ}\text{C}$  para cada uno de los 12 meses

hora	Ene	Feb	Mar	Abr	May	Jun	Jul	Ago	Sept	Oct	Nov	Dic
0	1,05	3,14	7,19	11,04	15,46	21,13	25,96	21,43	15,86	11,57	7,21	5,11
1	1,16	3,25	7,36	11,23	15,73	21,37	26,21	21,66	16,06	11,70	7,32	5,19
2	1,56	3,60	7,93	11,84	16,54	22,09	26,95	22,39	16,69	12,14	7,73	5,51
3	2,41	4,29	8,96	12,88	17,91	23,30	28,20	23,64	17,81	12,97	8,59	6,21
4	3,83	5,37	10,46	14,34	19,76	24,91	29,89	25,37	19,43	14,26	10,03	7,44
5	5,77	6,77	12,31	16,05	21,89	26,75	31,81	27,37	21,39	15,88	11,95	9,14
6	7,88	8,23	14,18	17,74	23,94	28,50	33,65	29,34	23,36	17,58	14,04	11,02
7	9,70	9,46	15,71	19,09	25,56	29,88	35,11	30,91	24,96	19,01	15,83	12,66
8	10,88	10,25	16,68	19,93	26,56	30,73	36,01	31,88	25,96	19,90	16,99	13,73
9	11,36	10,57	17,06	20,27	26,96	31,07	36,37	32,26	26,36	20,27	17,46	14,16
10	11,39	10,59	17,08	20,28	26,98	31,08	36,38	32,28	26,38	20,29	17,49	14,19
11	11,35	10,56	17,04	20,23	26,90	31,01	36,31	32,22	26,33	20,25	17,45	14,15
12	11,33	10,54	17,01	20,20	26,86	30,97	36,27	32,18	26,30	20,23	17,43	14,14
13	11,35	10,56	17,04	20,23	26,90	31,01	36,31	32,22	26,33	20,25	17,45	14,15
14	11,39	10,59	17,08	20,28	26,98	31,08	36,38	32,28	26,38	20,29	17,49	14,19
15	11,36	10,57	17,06	20,27	26,96	31,07	36,37	32,26	26,36	20,27	17,46	14,16
16	10,88	10,25	16,68	19,93	26,56	30,73	36,01	31,88	25,96	19,90	16,99	13,73
17	9,70	9,46	15,71	19,09	25,56	29,88	35,11	30,91	24,96	19,01	15,83	12,66
18	7,88	8,23	14,18	17,74	23,94	28,50	33,65	29,34	23,36	17,58	14,04	11,02
19	5,77	6,77	12,31	16,05	21,89	26,75	31,81	27,37	21,39	15,88	11,95	9,14
20	3,83	5,37	10,46	14,34	19,76	24,91	29,89	25,37	19,43	14,26	10,03	7,44
21	2,41	4,29	8,96	12,88	17,91	23,30	28,20	23,64	17,81	12,97	8,59	6,21
22	1,56	3,60	7,93	11,84	16,54	22,09	26,95	22,39	16,69	12,14	7,73	5,51
23	1,16	3,25	7,36	11,23	15,73	21,37	26,21	21,66	16,06	11,70	7,32	5,19

Cuadro 4.10: Perfil de temperaturas en °C en las coordenadas del emplazamiento según el método descrito en [10]

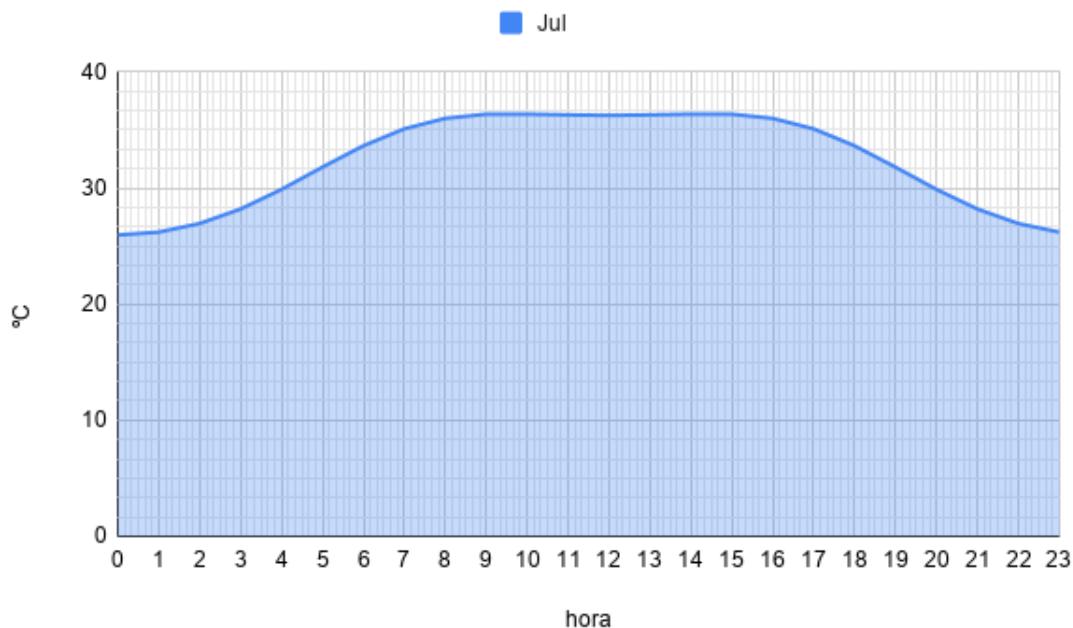


Figura 4.26: Perfil horario de temperaturas para el día promedio del mes de Julio

La temperatura ambiente se necesita para calcular la temperatura de la célula según la ecuación 3.32, que a su vez se utiliza para calcular la tensión de circuito abierto, como se puede ver en la ecuación 3.43.

A continuación, en la tabla 4.11 se muestran los resultados del proceso de cálculo descrito a partir de la página 31 para hallar la tensión de circuito abierto  $V_{oc}$ , la intensidad de cortocircuito  $I_{sc}$ , así como la tensión, intensidad y potencia en el punto de máxima potencia, todo ello en las condiciones de temperatura y radiación del emplazamiento indicado por el usuario

hora	$V_{oc}(V)$	$I_{sc}(A)$	$V_{mpp}(V)$	$I_{mpp}(A)$	$P_{mpp}(W)$	$P_{ac}(W)$
0	46,601	0	44,793	0	0	0
1	46,595	0	44,786	0	0	0
2	46,578	0	44,767	0	0	0
3	46,550	0	44,734	0	0	0
4	46,512	0	44,689	0	0	0
5	46,463	0,052	44,586	0,051	2,287	0
6	46,405	0,222	44,365	0,220	9,782	6,531
7	46,286	1,126	43,417	1,116	48,451	43,754
8	46,093	2,935	41,575	2,906	120,800	112,216
9	45,902	4,860	39,635	4,806	190,477	176,765
10	45,739	6,567	37,925	6,487	246,037	227,312
11	45,619	7,842	36,652	7,742	283,740	261,161
12	45,554	8,525	35,970	8,413	302,612	277,969
13	45,553	8,531	35,963	8,419	302,759	278,100
14	45,615	7,868	36,624	7,767	284,452	261,796
15	45,732	6,636	37,856	6,556	248,184	229,249
16	45,895	5,009	39,495	4,953	195,606	181,469
17	46,089	3,192	41,341	3,159	130,596	121,378
18	46,290	1,429	43,151	1,416	61,081	55,820
19	46,454	0,153	44,484	0,151	6,732	3,576
20	46,512	0	44,689	0	0	0
21	46,550	0	44,734	0	0	0
22	46,578	0	44,767	0	0	0
23	46,595	0	44,786	0	0	0

Cuadro 4.11: Valores horarios de  $V_{oc}$ ,  $I_{sc}$ ,  $I_{mpp}$ ,  $V_{mpp}$  y  $P_{mpp}$  para el día promedio del mes de julio

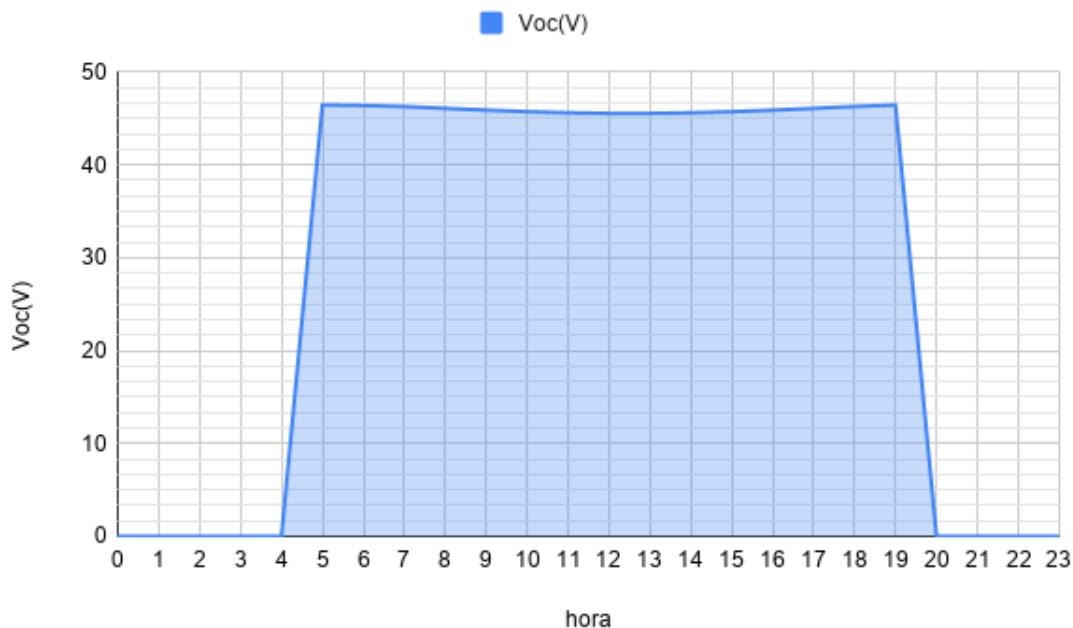


Figura 4.27: Valores horarios de tensión de circuito abierto para el día promedio del mes de Julio

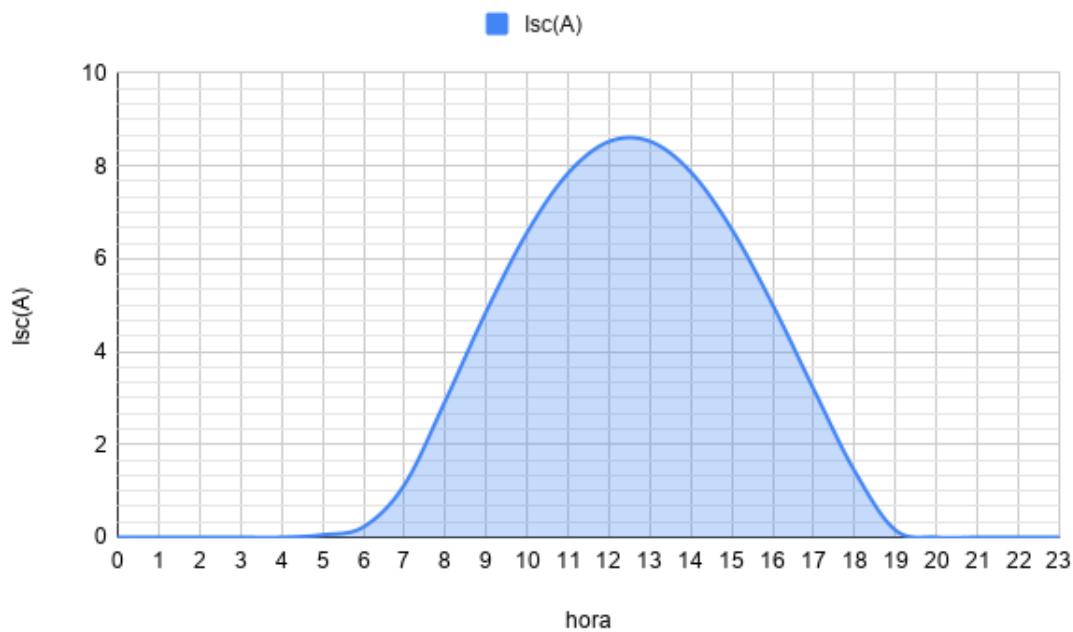


Figura 4.28: Valores horarios de corriente de cortocircuito para el día promedio del mes de Julio

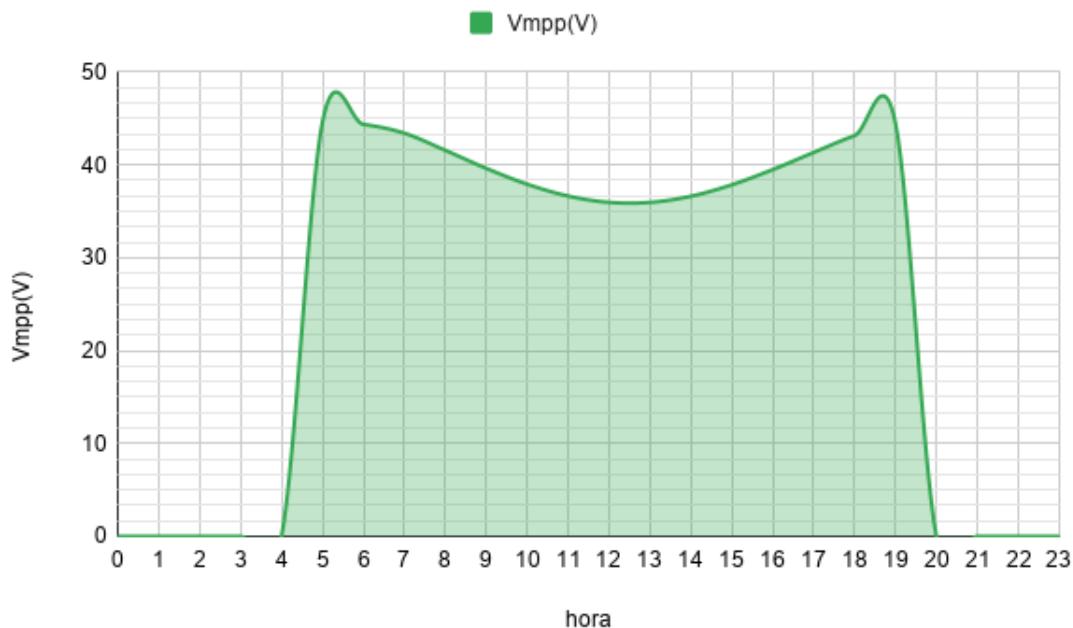


Figura 4.29: Valores horarios de tensión en el punto de máxima potencia para el día promedio del mes de Julio

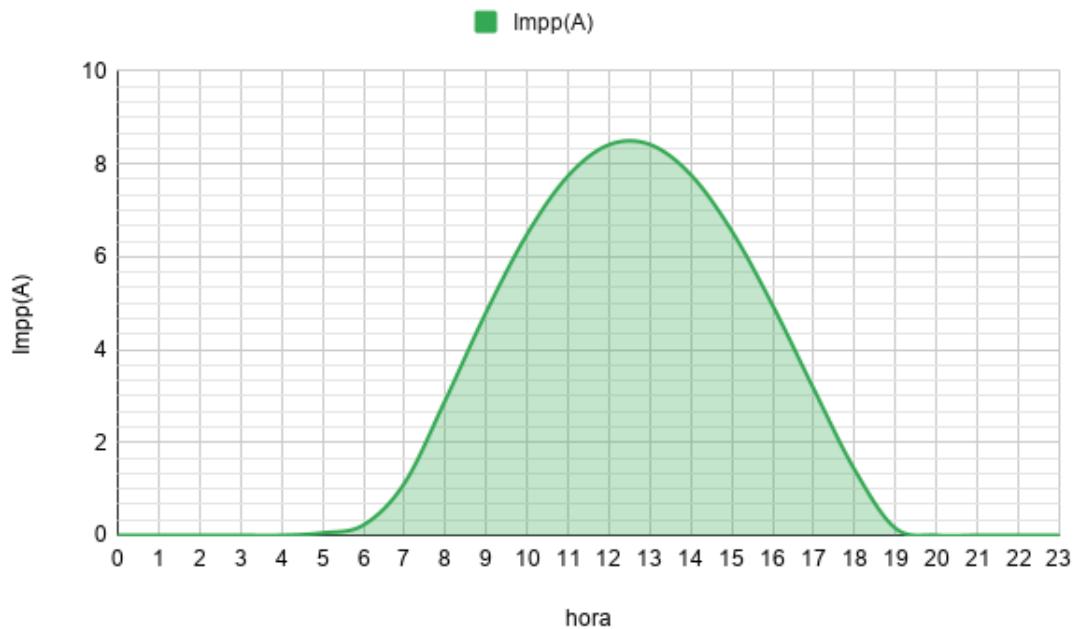


Figura 4.30: Valores horarios de corriente en el punto de máxima potencia para el día promedio del mes de Julio

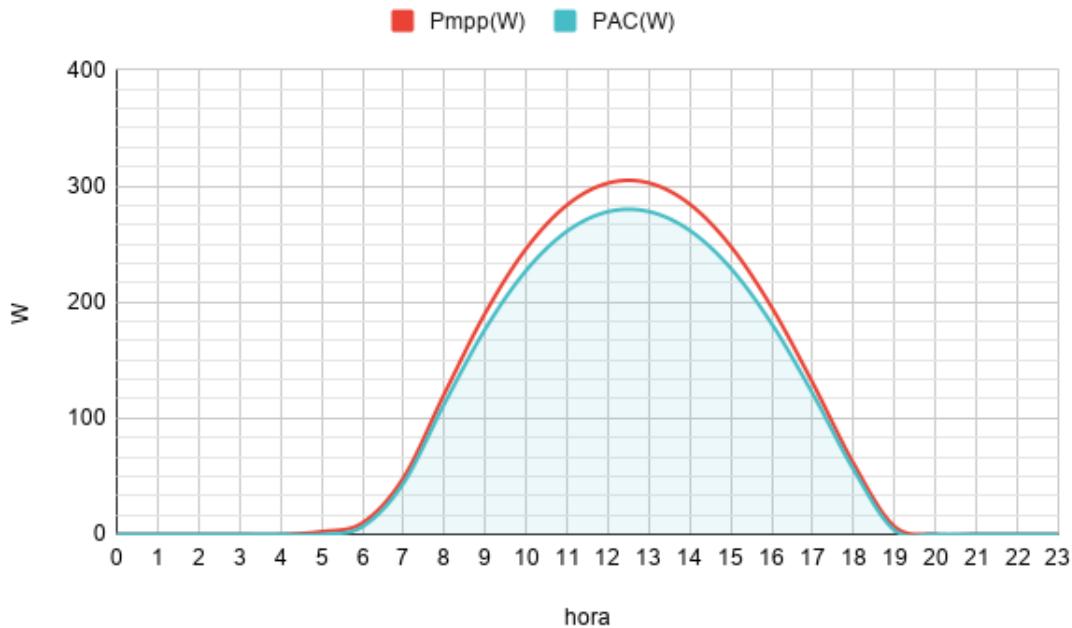


Figura 4.31: Valores horarios de potencia en el punto de máxima potencia y a la salida del inversor para el día promedio del mes de Julio

### 4.3. Cálculo final de las potencias y energías

Habiendo hecho un breve repaso por el proceso de cálculo en un emplazamiento concreto, lo último y más importante que debemos estudiar son los resultados finales que se han obtenido, es decir, las potencias máximas mensuales, la productividad y sobretodo la energía, tanto mensual como anual, que es capaz de producir el generador configurado por el usuario.

Hasta el momento, con el fin de simplificar los cálculos, éstos se llevaron a cabo con un solo módulo. A continuación vamos a trasladar todos estos valores a la superficie que el usuario va a destinar para la instalación del generador.

Para ello, debemos obtener una relación entre el área disponible que el usuario ha introducido y el área total del generador configurado (12 módulo en serie y 11 en paralelo). Para ello, utilizamos el número total de módulos del generador base multiplicado por el área del un módulo, que se indica como uno de los valores disponibles en la tabla 3.3.

En esta caso, la relación resultante es:

$$\begin{aligned}
 Area_{generador} &= 12 \cdot 11 \cdot 1,941m^2 = 256,257m^2 \\
 Area_{superficie} &= 40m^2 \\
 R &= \frac{Area_{superficie}}{Area_{generador}} = 0,156
 \end{aligned} \tag{4.1}$$

Por otro lado, podemos obtener el numero de módulos totales que se pueden instalar en el área indicada por el usuario simplemente dividiendo el área indicada por el área de un módulo y redondeando hacia abajo al primero numero entero. De tal manera que:

$$\begin{aligned}
 Area_{modulo} &= 1,941 m^2 \\
 Area_{superficie} &= 40 m^2 \\
 N_{modulos} &= \frac{40}{1,941} = 20,60 \simeq 20
 \end{aligned} \tag{4.2}$$

Por tanto, al conocer el numero de módulos, también podemos conocer la potencia nominal instalada, así como el resto de valores como la energía mensual, la potencia máxima que es capaz de entregar el generador o la productividad.

A continuación se muestran unos gráficos extraídos directamente de la página de resultados:



Figura 4.32: Energía mensual producida

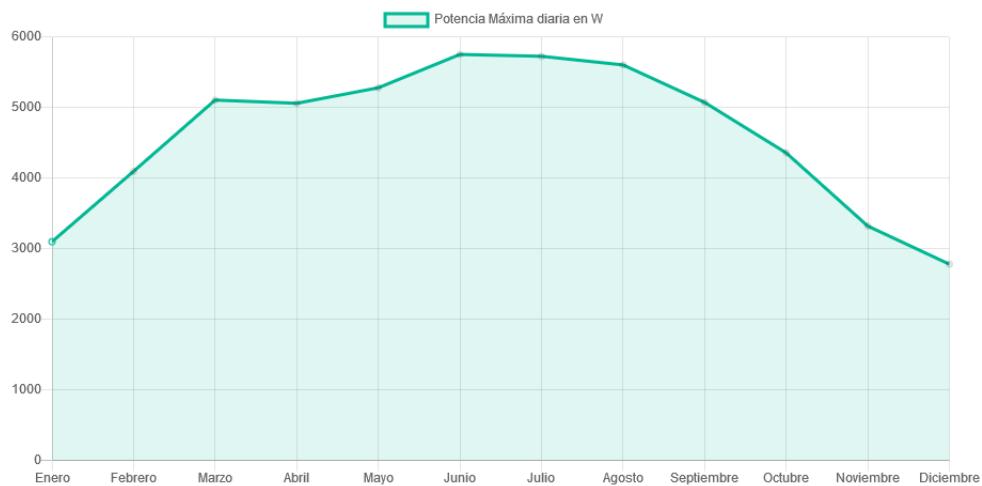


Figura 4.33: Potencia máxima mensual entregada

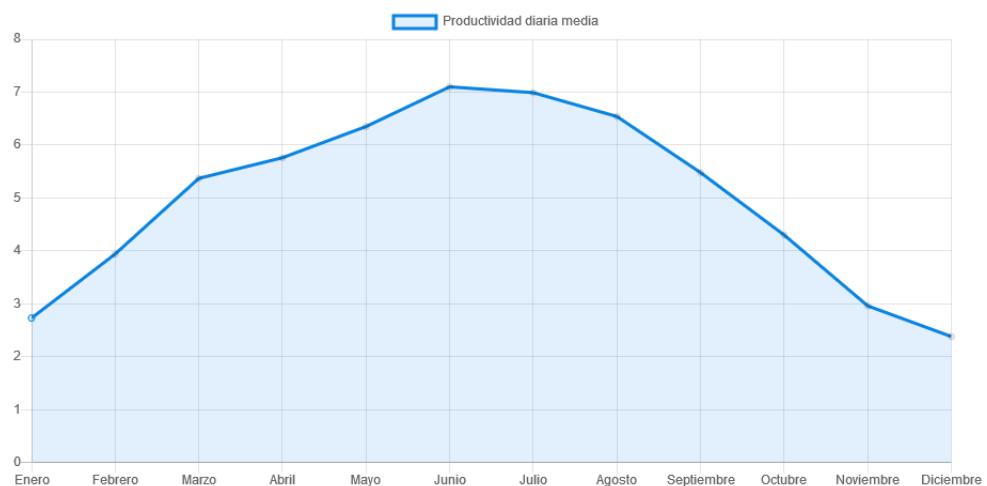


Figura 4.34: Productividad media mensual

## Capítulo 5

# Detalles de la programación

En este capítulo se va a llevar a cabo una explicación mas extensa de algunos de los apartados de la programación que hubieran resultado muy densos o complejos en la explicación del código. Además se hará una descripción más profunda de las tecnologías usadas en la aplicación. En este anexo solo se utilizarán los extractos de código que acompañen a una explicación o aclaración. Tanto el código completo de la aplicación como éste documento están almacenados en un repositorio online de la página de Github para un acceso rápido y cómodo a través del siguiente enlace: <https://github.com/IonutMorariu/PV-Calculator>

### 5.1. Descripción de tecnologías

En el desarrollo de cualquier aplicación web intervienen dos partes muy diferencias, cada una con unas funciones específicas.

Por un lado es necesario desarrollar un servidor, que en la industria se conoce como Backend, encargado de gestionar la base de datos donde se almacenan los datos, y proveer la información necesaria en todo momento a través de peticiones HTTP<sup>1</sup>.

Por otro lado, para que el usuario pueda interactuar con la página, es necesario desarrollar una interfaz para la web, esto se conoce como Frontend o cliente y es lo que se encarga de recoger la información del usuario a través de un formulario que es posteriormente enviado al servidor. Éste realiza los cálculos necesarios, en este caso todo lo relacionado con la radiación solar, y los devuelve al cliente para mostrar los resultados al usuario.

#### 5.1.1. Servidor

En el caso del servidor, existe una gran variedad de lenguajes de programación que se pueden utilizar, cada uno con sus ventajas y desventajas.

Los lenguajes más utilizados para la programación de servidores web son:

- **Java:** Es quizás el lenguaje de programación de servidores más extendido y usado en la industria

---

<sup>1</sup>HTTP: Protocolo de comunicación para transmitir información a través de la web

- **Python:** Un lenguaje de programación que destaca por su semejanza con el lenguaje natural. Ha adquirido bastante fama en los últimos años en temas relacionados con el aprendizaje máquina
- **PHP:** Uno de los lenguajes más extendidos, en parte por ser el que se ha utilizado para programar Wordpress, el gestor de contenido más utilizado.
- **Javascript:** Un lenguaje creado para programar páginas web, pero que gracias a un entorno de ejecución desarrollado por Google, se puede utilizar para crear servidores. La popularidad de este lenguaje viene de la flexibilidad que ofrece, dado que un solo programador puede realizar las dos tareas, servidor y cliente, si conoce este lenguaje. Comúnmente es confundido con Java, pero son totalmente diferentes.

El lenguaje escogido en este caso es Javascript, por experiencias anteriores y por la flexibilidad de utilizar un solo lenguaje en ambos casos. Es un lenguaje de programación interpretado, es decir, no es necesario compilarlo con cada cambio, sino que se ejecuta directamente desde el archivo de código, agilizando el desarrollo a riesgos de cometer más errores en vivo.

El entorno de Javascript que nos permite crear el servidor es NodeJS. Según la descripción de la página web [11], NodeJS fue ideado como un entorno de ejecución de Javascript orientado a eventos asíncronos, diseñado para crear aplicaciones de red escalables.

Para la base de datos también existen multitud de opciones con diferentes ventajas y desventajas. Algunas de las opciones más utilizadas son:

- **MySQL:** La base de datos relacional más extendida, aunque por su edad y arquitectura de diseño, carece de muchas funcionalidades necesarias en el estado actual de la web.
- **PostgreSQL:** Una base de datos relacional más moderna, y con mas funcionalidad que MySQL, que está en proceso de sustituir.
- **MongoDB:** A diferencia de las SQL, MongoDB es una base de datos no relacional, que guarda la información en documentos individuales en lugar de tablas.

La similitud entre la sintaxis de las peticiones y los resultados de MongoDB con la de Javascript ha hecho que sea la opción elegida para este proyecto.

```

1   db.inventory.find( {
2     status: "A",
3     $or: [ { qty: { $lt: 30 } }, { item: /^p/ } ]
4   )

```

Extracto de código 5.1: Ejemplo de una petición en MongoDB

Se puede observar que tiene muchas similitudes con el lenguaje de programación utilizado, a diferencia de SQL que tiene una sintaxis mas particular:

```

1   SELECT * FROM inventory WHERE status = "A" AND ( qty < 30 OR item LIKE "p%")

```

Extracto de código 5.2: Ejemplo de una petición en SQL

### 5.1.2. Cliente

Al el contrario que en el servidor, en el cliente no existen opciones a la hora de programar la interfaz. Solamente se pueden usar 3 lenguajes y cada uno de ellos tiene una función específica.

#### HTML

HTML es el lenguaje que se utiliza para establecer la estructura de la página y dotarla de contenido. Es decir, es lo que se utiliza para definir que campos tendrá el formulario, el título de la página y cualquier otro tipo de contenido.

La sintaxis de este lenguaje se conoce como lenguaje de marcado, de ahí sus siglas (Hyper Text Markup Language) que se basa en una etiquetas concretas que define sus estructura.

```

1  <!DOCTYPE html>
2
3  <html>
4      <head>
5          <title>Page Title</title>
6      </head>
7      <body>
8          <h1>Esto es un titulo</h1>
9          <p>This is a parágrafo.</p>
10         </body>
11     </html>
```

Extracto de código 5.3: Ejemplo de una petición en SQL

#### CSS

CSS (Cascading Style Sheet) es el lenguaje que se utiliza en una web para proveer de estilo al contenido definido con HTML. Utiliza un sistema de propiedades y valores que heredan de más general a más específico.

Al tratarse de simplemente listas de propiedades que se aplican a las diferentes etiquetas de HTML, es bastante común utilizar códigos ya creados para ahorrar tiempo. Estas listas son también conocidas como librerías de componentes. Las más conocidas son Bootstrap<sup>2</sup>, Bulma<sup>3</sup> y Zurb Foundation<sup>4</sup>

```

1  body {
2      background-color: lightblue;
3  }
```

<sup>2</sup>Bootstrap: <https://getbootstrap.com/>

<sup>3</sup>Bulma: <https://bulma.io/>

<sup>4</sup>Zurb Foundation: <https://get.foundation/>

```

5   h1 {
6       color: white;
7       text-align: center;
8   }
9
10  p {
11      font-family: verdana;
12      font-size: 20px;
13 }
```

#### Extracto de código 5.4: Ejemplo de código CSS

La forma en la que se escribe dicho código es aplicando ciertas reglas u normas de estilo predefinidas a las diferentes etiquetas de HTML. Así, por ejemplo, según el extracto 14 todo el documento tendrá de color de fondo el azul, o el título de la página será verde y alineado al centro.

### Javascript

Como ya se ha mencionado anteriormente, Javascript es el lenguaje de programación utilizado tanto para el servidor como para el cliente.

En el caso del cliente no existen otras opciones dado que Javascript es el único lenguaje que los navegadores web utilizados en la actualidad son capaces de interpretar.

A diferencia de HTML y CSS, Javascript es el encargado de otorgarle la lógica y la funcionalidad a la página web, pues es el que realiza la recogida de datos y su posterior envío al servidor para que se lleven a cabo parte de los cálculos.

En el cliente se realizan los cálculos relacionados con la potencia y la energía entregadas, pues no requiere de acceso a la base de datos y por tanto es posible delegar parte de la carga de trabajo que soporta el servidor.

## 5.2. Explicaciones de código

### 5.2.1. Emplazamiento del usuario

Uno de los datos más importantes para poder estimar la energía que finalmente podrá producir el generador es el de la radiación solar en el lugar de instalación. Para ello, es necesario conocer las coordenadas geográficas del emplazamiento. Sin embargo, es poco intuitivo pedirle a un usuario que introduzca sus coordenadas, dado que la mayoría desconocen dichos datos.

Por lo tanto, la ruta que se ha tomado es la de pedirle al usuario su dirección, o una dirección cercana a su localización, y utilizar la API de Google Maps<sup>5</sup> para convertir dicha dirección en las coordenadas de latitud y longitud que se necesitan para poder obtener la radiación en dicho lugar.

---

<sup>5</sup>API Google Maps: Enlace interactivo al que se le pueden enviar los datos de una dirección y devuelve las coordenadas de latitud y longitud de un emplazamiento.

Este proceso comienza por recoger los datos de la dirección, municipio y código postal a través del formulario que aparece en la página web.

Estos datos son recogidos en el código a través de un nombre único que han recibido:

```
1 const addressInput = document.querySelector('#address');
2 const cityInput = document.querySelector('#city');
3 const postalInput = document.querySelector('#postal');
```

Extracto de código 5.5: Variables correspondientes a los tres campos

Una vez que tenemos estos datos recogidos en las variables, podemos pedir a la API de Google Maps las coordenadas de latitud y longitud de dicho emplazamiento encadenando las tres variables y una clave única de identificación, para obtener un enlace único que se corresponde a dicha localización.

```
1 const getCoordinates = async () => {
2   const address = addressInput.value.split(' ').join('+');
3   const city = cityInput.value;
4   const postal = postalInput.value;
5   const requestURL = `${googleEndpoint}address=${address},${city},${postal}
6                           ,spain&key=${googleApiKey}`;
7   const response = await fetch(requestURL);
8   const data = await response.json();
9   const info = {
10     formattedAddress: data.results[0].formatted_address,
11     lat: data.results[0].geometry.location.lat,
12     long: data.results[0].geometry.location.lng
13   };
14
15   return info;
16};
```

Extracto de código 5.6: Función encargada de solicitar los datos a la API

La función de **getCoordinates** (5.6) recoge el valor de la dirección y reemplaza los espacios con el signo + (formato requerido por la API) y lo concatena con el valor del campo de la ciudad y el código postal. Al final le añade una clave única que identifica la aplicación a la hora de establecer límites de uso y evitar abuso de la API.

Una vez creado este enlace único, el código lanza la petición al servicio y retorna con la información que es recogida y se guarda en dos variables **lat** y **long** para ser utilizadas posteriormente, a la hora de obtener los datos de irradiación global.

### 5.2.2. Área, inclinación, orientación y nivel de suciedad de la superficie de instalación

Además de las información de latitud y longitud del emplazamiento, el cálculo de la instalación también requiere de información relacionada con el área, la inclinación, la orientación y el nivel de suciedad de la superficie donde se va a realizar la instalación, para poder realizar una estimación lo mas exacta posible.

Estos valores son recogidos directamente de los campos de la pagina web, al igual que los campos anteriores, sin necesitar ningún trato especial:

```

1 const slope = document.querySelector('#slope');
2 const area = document.querySelector('#area');
3 const orientation = document.querySelector('#orientation');
4 const dirtLevel = document.querySelector('#dirt-level');
```

Extracto de código 5.7: Variables correspondientes a los campos indicados

### 5.2.3. Obtención de la información de radiación global en el plano horizontal

En la sección teórica se menciona brevemente como se obtuvo la radiación global en el plano horizontal de manera dinámica en función de la latitud y longitud del emplazamiento.

En ésta sección se va a llevar a cabo un desarrollo más extenso del proceso que se llevó a cabo para poder tener acceso a la información.

El proceso comienza con la búsqueda de una fuente de la radiación global. Investigando las diferentes posibilidades, surge la página de ADRASE, un proyecto fundado por el Gruo de Radiación Solar del CIEMAT. En la página web<sup>6</sup> se puede encontrar una descripción detallada acerca del proyecto.

En la página de ADRASE podemos encontrar un mapa interactivo de la península, en la que haciendo clic en cualquier punto, nos aparece un enlace con los datos mensuales de radiación, como se puede observar en la figura 5.1.

Sin embargo, éste mapa no permite tener un acceso dinámico a la información, sin la interacción de un usuario. Además el servicio de ADRASE no ofrece un punto de acceso a los datos, por tanto, la descarga de datos se tiene que hacer previamente y almacenar los datos en la base de datos del servidor.

Cuando se hace clic en el enlace de datos mensuales de radiación global, aparece una ventana emergente con los datos del emplazamiento, como el de la figura 5.2.

La ventana emergente tiene un enlace personalizado de tal manera que pueda generar la información de ese localización concreta. En enlace contiene dos variables que hacen referencia a la latitud y la longitud:

```
1 www.adrase.com/adrasemaps/php/monthly_popup.php?lat=39.12&lon=-3.92&var_tipe=0
```

Extracto de código 5.8: Enlace tipo de la radiación en la peninsula

Por tanto, si generamos un enlace con la latitud y la longitud del emplazamiento, podemos obtener una ventana emergente como la que se muestra en la figura 5.2. Sin embargo, obtener los datos

<sup>6</sup>ADRASE: <http://www.adrase.com/el-proyecto.html>

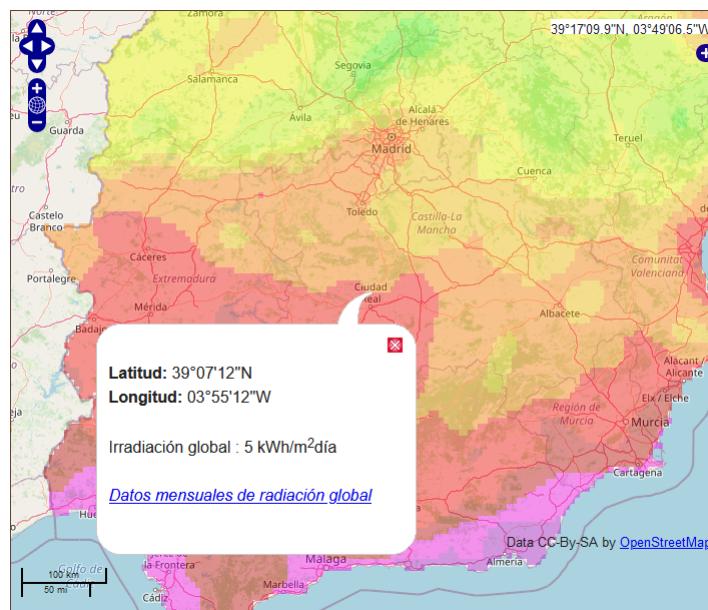


Figura 5.1: Mapa interactivo de radiación

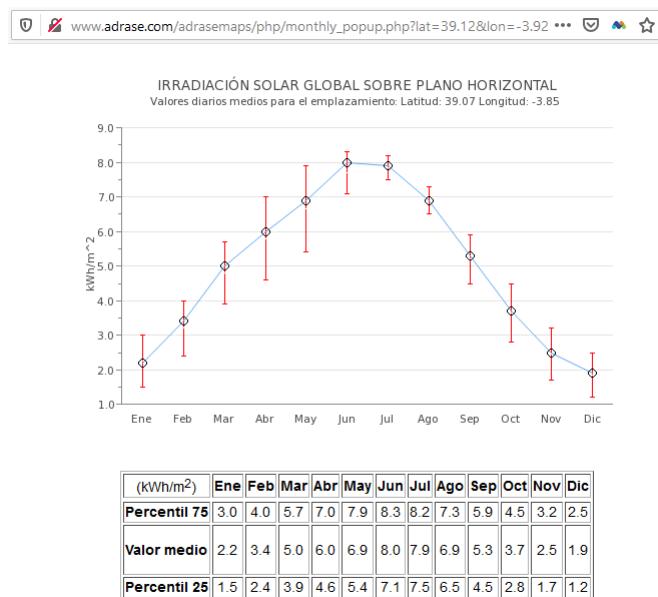


Figura 5.2: Ventana emergente

de radiación no es una tarea inmediata. Es necesario procesar el documento y extraer los valores deseados.

El contenido del documento contiene una tabla de html que contiene los valores medios, percentil 25 y percentil 75 de radiación global en el plano horizontal. La parte que necesitamos extraer y procesar es la siguiente:

```

1 <tr class="customPeriods">
2     <td align="center"><b>Valor medio</b></td>
3     <td align="center"><p>2.2</p></td>
4     <td align="center"><p>3.3</p></td>
5     <td align="center"><p>4.8</p></td>
6     <td align="center"><p>6.0</p></td>
7     <td align="center"><p>7.0</p></td>
8     <td align="center"><p>8.0</p></td>
9     <td align="center"><p>7.8</p></td>
10    <td align="center"><p>6.9</p></td>
11    <td align="center"><p>5.3</p></td>
12    <td align="center"><p>3.7</p></td>
13    <td align="center"><p>2.4</p></td>
14    <td align="center"><p>1.9</p></td>
15 </tr>
```

Extracto de código 5.9: Tabla de valores medios según documento de ADRASE

Por tanto, para procesar el documento, lo que haremos es buscar la frase *Valor medio* y eliminar todo lo anterior. A continuación, buscaremos en el documento el primer *</tr>* que se encuentra, dado que indicará que se ha terminado la fila, ya que la etiqueta *<tr>* de HTML significa table row.

Una vez aisladas las 12 líneas de código que contienen los 12 valores que nos interesa, usaremos una expresión regular (RegEx<sup>7</sup>) para aislar solamente la parte numérica de cada fila.

En código resulta:

```

1 let texts = [];
2 if (e.includes('NO HAY DATOS PARA EL PUNTO SELECCIONADO')) {
3     texts.push(null);
4 } else if (e.includes('Valor medio')) {
5     const regex = /[+-]?\d+(\.\d+)?/g;
6     const str = e.split('Valor medio')[1].split('tr')[0];
7     const floats = str.match(regex).map(function(v) {
8         return parseFloat(v);
```

<sup>7</sup>RegEx: secuencia de caracteres que conforma un patrón de búsqueda

```

9   });
10  texts.push(floats);
11 }

```

#### Extracto de código 5.10: Proceso de extracción de los valores

En el extracto de código 12 se observa, en la primera linea de código, una condición tal que si el documento contiene la frase *NO HAY DATOS PARA EL PUNTO SELECCIONADO*, se evite el proceso de cálculo, ya que ocasiones, para una coordenada en concreto, sobretodo cercanas a la costa, ADRASE no es capaz de entregarnos los datos de radiación.

Sin embargo, si esa frase no existe, significa que sí existen dichos datos y por tanto podemos extraerlos según el proceso detallado.

La función *split()* es la que se ocupa de cortar el documento por el valor indicado, por tanto, como se observa, la utilizamos para cortar por *Valor medio* y por la etiqueta *tr*, quedando solamente las doce filas de valores medios.

Habiendo creado un script capaz de extraer los doce valores necesarios del documento, podemos replicar este proceso todas las veces que sea necesario.

Por tanto, el siguiente paso es llevar a cabo el proceso para una matriz de latitudes y longitudes, y guardar la información en la base de datos. Para ello, se generará un listado de enlaces al que aplicaremos el algoritmo anterior.

El código que utilizaremos para generar los enlaces es:

```

1  const minLat = 36.1;
2  const maxLat = 43.63;
3  const minLon = -9.15;
4  const maxLon = 3.06;
5  const links = [];
6  const latDiff = 0.25;
7  const lonDiff = 0.25;
8  for (let i = minLat; i <= maxLat; i += 0.25) {
9    for (let j = minLon; j <= maxLon; j += 0.25) {
10      const lon = j.toFixed(2);
11      const lat = i.toFixed(2);
12      links.push(
13        `http://www.adrase.com/adrasmaps/php/monthly_popup.php?
14        lat=${lat}&lon=${lon}&var_tipe=0`
15      );
16    }
17  }

```

#### Extracto de código 5.11: Proceso de generación de enlaces

Las variables que influyen en el bucle de generación de enlaces son:

- **minLat:** Latitud a partir de la cual se comienza a generar los enlaces.
- **maxLat:** Valor máximo de latitud para parar el bucle.
- **minLat:** Longitud a partir de la cual se comienza a generar los enlaces.
- **maxLat:** Valor máximo de longitud para parar el bucle.
- **links:** Un vector o lista para almacenar los enlaces.
- **latDiff:** Diferencia de latitud en grados entre un punto y el siguiente.
- **lonDiff:** Diferencia de longitud en grados entre un punto y el siguiente.

Con estos valores, lo que conseguimos es configurar una cuadricula con un tamaño concreto, de la que generamos los enlaces. En el caso de la configuración expuesta en el ejemplo, el numero de enlaces que se genera es de 860. A cada uno de los elementos de la lista le aplicaremos el algoritmo de extracción de datos, y obtendremos un documento JSON<sup>8</sup> con el siguiente aspecto:

```

1  {
2      "lat":"36.35",
3      "lon": "-6.15",
4      "midValues": [
5          2.7,
6          3.8,
7          5.2,
8          6.2,
9          7.3,
10         7.7,
11         7.8,
12         7,
13         5.8,
14         4.4,
15         3.2,
16         2.5
17     ]
18 },

```

Extracto de código 5.12: Ejemplo de un documento JSON con la información de un punto concreto

---

<sup>8</sup>JSON: Javascript Object Notation es una forma de codificar información, similar a un CSV, pero diseñado específicamente para trabajar de manera nativa con Javascript

Cada uno de los documentos generados es mandado al servidor y guardado en la base de datos. Posteriormente, utilizando una función nativa de MongoDB podremos encontrar el punto más cercano a las coordenadas que envía el usuario, y por tanto obtener una aproximación de la radiación el su emplazamiento.

# Apéndice A

## Código completo

Todo el código que se muestra a continuación está disponible para descargar, consultar y modificar libremente en el repositorio de Github correspondiente a este trabajo a través del enlace: <https://github.com/IonutMorariu/PV-Calculator>

### A.1. Código del servidor

#### A.1.1. DataController.JS

```
1 const path = require('path');
2 const mongoose = require('mongoose');
3 const SolarData = mongoose.model('SolarData');
4 const StationData = mongoose.model('StationData');
5 const helpers = require('../helpers/helpers');
6 mongoose.Promise = global.Promise;
7 const axios = require('axios');

8
9 deg2rad = (degs) => {
10   return (degs * Math.PI) / 180;
11 };
12 rad2deg = (rads) => {
13   return (rads * 180) / Math.PI;
14 };
15
16 exports.getData = async (req, res, next) => {
17   const latitude = parseFloat(req.query.latitude);
```

```
18 const longitude = parseFloat(req.query.longitude);
19 const location = await SolarData.find({
20   location: {
21     $near: {
22       $maxDistance: 10000,
23       $geometry: {
24         type: 'Point',
25         coordinates: [longitude, latitude]
26       }
27     }
28   }
29 });
30 const locationData = location[0];
31 res.json({ data: locationData });
32 };
33 exports.saveData = async (req, res, next) => {
34   const data = require('./data.json');
35   data.forEach(async (elem) => {
36     const data = await new SolarData({
37       location: {
38         type: 'Point',
39         coordinates: [parseFloat(elem.lon), parseFloat(elem.lat)]
40       },
41       meanValues: elem.midValues
42     }).save();
43   });
44   res.json({ Data: 'Arrived' });
45 };
46
47 exports.doCalculations = async (req, res, next) => {
48   if (req.query.longitude == undefined || req.query.latitude == undefined) {
49     res.status(400).send('Longitude or latitude is undefined');
50   }
51   const latitude = parseFloat(req.query.latitude);
52   const longitude = parseFloat(req.query.longitude);
53   const angle = parseFloat(req.query.angle);
```

```
54 const applyDirtLevel = req.query.applyDirtLevel;
55 const dirtLevel = req.query.dirtLevel;
56 const orientation = parseFloat(req.query.orientation);
57 const location = await SolarData.find({
58   location: {
59     $near: {
60       $maxDistance: 25000,
61       $geometry: {
62         type: 'Point',
63         coordinates: [longitude, latitude]
64       }
65     }
66   }
67 });
68 if (location == []) {
69   res.status(204);
70   res.send('No data found for this location.
71 Please make sure you have entered a valid pair of coordinates');
72   return;
73 }
74 const locationData = location[0];
75
76 const calcData = {
77   longitude: locationData.location.coordinates[0],
78   latitude: locationData.location.coordinates[1],
79   angle,
80   orientation,
81   applyDirtLevel,
82   dirtLevel,
83   meanValues: [
84     {
85       month: 'Jan',
86       normalDay: 17,
87       meanGR: locationData.meanValues[0]
88     },
89     {

```

```
90     month: 'Feb',
91     normalDay: 45,
92     meanGR: locationData.meanValues[1]
93   },
94   {
95     month: 'Mar',
96     normalDay: 74,
97     meanGR: locationData.meanValues[2]
98   },
99   {
100    month: 'Apr',
101    normalDay: 105,
102    meanGR: locationData.meanValues[3]
103  },
104  {
105    month: 'May',
106    normalDay: 135,
107    meanGR: locationData.meanValues[4]
108  },
109  {
110    month: 'Jun',
111    normalDay: 161,
112    meanGR: locationData.meanValues[5]
113  },
114  {
115    month: 'Jul',
116    normalDay: 199,
117    meanGR: locationData.meanValues[6]
118  },
119  {
120    month: 'Aug',
121    normalDay: 230,
122    meanGR: locationData.meanValues[7]
123  },
124  {
125    month: 'Sep',
```

```
126     normalDay: 261,
127     meanGR: locationData.meanValues[8]
128   },
129   {
130     month: 'Oct',
131     normalDay: 292,
132     meanGR: locationData.meanValues[9]
133   },
134   {
135     month: 'Nov',
136     normalDay: 322,
137     meanGR: locationData.meanValues[10]
138   },
139   {
140     month: 'Dic',
141     normalDay: 347,
142     meanGR: locationData.meanValues[11]
143   }
144 ]
145 };
146
147 const resultData = helpers.calculateValues(calcData);
148
149 res.json({ data: resultData });
150 };
151
152 exports.saveStations = async (req, res, next) => {
153   const response = await axios(
154     'https://opendata.aemet.es/opendata/api/valores
155     /climatologicos/inventarioestaciones/todasestaciones/?api_key='
156     + process.env.AEMET_API
157   );
158   const data = response.data;
159   const datosResponse = await axios(data.datos);
160   const estaciones = datosResponse.data;
161   const convertedStations = estaciones.map((station) => {
```

```
162     const latLetter = station.latitud[station.latitud.length - 1];
163     const lonLetter = station.longitud[station.longitud.length - 1];
164     const latSign = latLetter == 'N' ? 1 : -1;
165     const lonSign = lonLetter == 'W' ? -1 : 1;
166     const lat = latSign * (parseFloat(station.latitud.substr(0,
167         station.latitud.length - 1)) / 10000);
168     const lon = lonSign * (parseFloat(station.longitud.substr(0,
169         station.longitud.length - 1)) / 10000);
170     const stationObj = {
171         emaid: station.indicativo,
172         lat,
173         lon
174     };
175     return stationObj;
176 });
177 convertedStations.forEach(async (elem) => {
178     const data = await new StationData({
179         location: {
180             type: 'Point',
181             coordinates: [elem.lon, elem.lat]
182         },
183         emaid: elem.emaid
184     }).save();
185 });
186 res.json({ Data: 'Arrived' });
187 };
188
189 exports.getTemperatureProfile = async (req, res, next) => {
190     const latitude = parseFloat(req.query.latitude);
191     const longitude = parseFloat(req.query.longitude);
192     let response;
193     const stations = await StationData.find({
194         location: {
195             $near: {
196                 $maxDistance: 50000,
197                 $geometry: {
```

```
198     type: 'Point',
199     coordinates: [longitude, latitude]
200   }
201 }
202 }
203 });
204 const monthlyValues = [];
205 stations.forEach(async (station, index) => {
206   const { emaId } = station;
207   const aemetLink = `https://opendata.aemet.es/
208   opendata/api/valores/climatologicos/mensualesanuales/datos
209   /anioini/2015/aniofin/2015/estacion/${emaId}/?api_key=${
210     process.env.AEMET_API
211   }`;
212   const firstRes = await axios(aemetLink);
213   const firstData = firstRes.data;
214   if (firstData.estado == 404) {
215     return;
216   } else if (firstData.estado == 200) {
217     const secondRes = await axios(firstData.datos);
218     const secondData = secondRes.data;
219     monthlyValues.push(secondData);
220   }
221   if (index == stations.length - 1) {
222     const tempArray = monthlyValues[0].map((month) => {
223       const monthNumber = month.fecha.split('-')[1];
224       const Tmin = parseFloat(month.tm_min) || 5;
225       const Tmax = parseFloat(month.tm_max) || 20;
226       return { Tmin, Tmax, monthNumber };
227     });
228     const normalDays = [17, 45, 74, 105, 135, 161, 199, 230, 261, 292, 322, 347];
229
230     const wp = Math.PI / 4;
231
232     tempArray.pop();
233   }
```

```

234     const tempProfiles = tempArray.map((elem, index) => {
235         const { Tmax, Tmin, monthNumber } = elem;
236         const Tm = (Tmax + Tmin) / 2;
237         const Tr = (Tmax - Tmin) / 2;
238         const Ta = [];
239         const decl = 23.45 * Math.sin((2 * Math.PI * (normalDays[index] + 284)) / 365);
240         const cosWs = -Math.tan(deg2rad(decl)) * Math.tan(deg2rad(latitude));
241         const ws = -Math.acos(cosWs);
242         for (let h = -12; h < 12; h++) {
243             const w = Math.cos(deg2rad(h * 15));
244             const a1 = (Math.PI * 12 * (ws - w)) / (21 * Math.PI + 12 * ws);
245             const a2 = (Math.PI * (3 * Math.PI - 12 * w)) / (3 * Math.PI - 12 * ws);
246             const a3 = (Math.PI * (24 * Math.PI + 12 * (ws - w))) /
247             (21 * Math.PI + 12 * ws);
248             const T1 = Tm - Tr * Math.cos(a1);
249             const T2 = Tm + Tr * Math.cos(a2);
250             const T3 = Tm - Tr * Math.cos(a3);
251             if (w <= ws) {
252                 Ta.push(T1);
253             } else if (w > ws && w <= wp) {
254                 Ta.push(T2);
255             } else if (w > wp) {
256                 Ta.push(T3);
257             }
258         }
259         return { hourlyTa: Ta, monthNumber, Tmax, Tmin };
260     });
261     response = { latitude, longitude, profiles: tempProfiles };
262     if (response) {
263         res.json(response);
264     } else {
265         res.json({ Data: 'Error with data' });
266     }
267 }
268 );
269 };

```

## Extracto de código A.1: Server/Controllers/dataController.js

### A.1.2. Helpers.JS

```
1 deg2rad = (degs) => {
2     return (degs * Math.PI) / 180
3 }
4 rad2deg = (rads) => {
5     return (rads * 180) / Math.PI
6 }
7
8 const cos = (rads) => {
9     return Math.cos(rads)
10 }
11
12 const sin = (rads) => {
13     return Math.sin(rads)
14 }
15 exports.calculateValues = (data) => {
16     const newData = data
17     //1. Declinación, excentricidad y ángulo amanecer
18     newData.meanValues.forEach((elem, index) => {
19         const decl = 23.45 * Math.sin((2 * Math.PI * (elem.normalDay + 284)) / 365)
20         const exct = 1 + 0.033 * Math.cos((2 * Math.PI * elem.normalDay) / 365)
21         elem.decl = decl
22         elem.exct = exct
23         const cosWs = -Math.tan(deg2rad(decl)) * Math.tan(deg2rad(newData.latitude))
24         const ws = -Math.acos(cosWs)
25         elem.ws = ws
26     })
27     //2. Cálculo Bo
28     newData.meanValues.forEach((elem, index) => {
29         const Bo = 1.367
30         const zenith = []
```

```
31 const tilt = []
32 const B00 = []
33 for (let h = -12; h < 12; h++) {
34     const cosZenit =
35         Math.cos(deg2rad(elem.decl)) * Math.cos(deg2rad(h * 15)) *
36         Math.cos(deg2rad(newData.latitude)) +
37         Math.sin(deg2rad(elem.decl)) * Math.sin(deg2rad(newData.latitude))
38     const zenithVal = rad2deg(Math.acos(cosZenit))
39     zenith.push(zenithVal)
40 }
41 elem.zenit = zenith
42 for (let h = -12; h < 12; h++) {
43     const betaRad = deg2rad(newData.angle)
44     const alphaRad = deg2rad(newData.orientation)
45     const hRad = deg2rad(h * 15)
46     const latRad = deg2rad(newData.latitude)
47     const declRad = deg2rad(elem.decl)
48     const anglIncidenciaVal =
49         Math.sign(newData.latitude) *
50         (Math.sin(betaRad) * Math.cos(alphaRad) * Math.cos(declRad) *
51         Math.cos(hRad) * Math.sin(latRad) -
52         Math.sin(betaRad) * Math.cos(alphaRad) * Math.cos(latRad) *
53         Math.sin(declRad)) +
54         Math.sin(betaRad) * Math.sin(alphaRad) * Math.cos(declRad) * Math.sin(hRad) +
55         Math.cos(betaRad) * Math.cos(declRad) * Math.cos(hRad) * Math.cos(latRad) +
56         Math.cos(betaRad) * Math.sin(declRad) * Math.sin(latRad)
57     console.log({
58         data: { angle: newData.angle, orientation: newData.orientation,
59             lat: newData.latitude, decl: elem.decl },
60         h,
61         anglIncidenciaVal,
62         mes: index,
63     })
64     const anglIncidencia = rad2deg(Math.acos(anglIncidenciaVal))
65     tilt.push(anglIncidencia)
66     elem.tilt = tilt
```

```
67    }
68
69    for (let h = 0; h < 24; h++) {
70      const B00val = Bo * elem.exct * Math.cos(deg2rad(elem.zenit[h]))
71      B00.push(B00val)
72    }
73    elem.B00 = B00
74
75    const bod =
76      -(24 / Math.PI) *
77      Bo *
78      elem.exct *
79      (elem.ws * Math.sin(deg2rad(newData.latitude)) *
80       Math.sin(deg2rad(elem.decl)) +
81       Math.cos(deg2rad(elem.decl)) * Math.cos(deg2rad(newData.latitude)))
82      * Math.sin(elem.ws))
83    elem.B0d0 = bod
84  })
85 //3. Calculo del indice de claridad y Fd
86 newData.meanValues.forEach((elem, index) => {
87   const Ktd = elem.meanGR / elem.B0d0
88   elem.Ktd = Ktd
89   const Fd = 1 - 1.13 * elem.Ktd
90   elem.Fd = Fd
91 })
92
93 //4. Calculo de Dd y Bd
94 newData.meanValues.forEach((elem, index) => {
95   const Dd0 = elem.Fd * elem.meanGR
96   const Bd0 = elem.meanGR - Dd0
97
98   elem.Dd0 = Dd0
99   elem.Bd0 = Bd0
100 })
101
102 //5. Calculo de la ecuacion de perfil (rd & rg)
```

```

103  newData.meanValues.forEach((elem, index) => {
104    const rd = []
105    for (let h = -12; h < 12; h++) {
106      const hRad = Math.cos(deg2rad(h * 15))
107      const rdval = (Math.PI / 24) * ((hRad - Math.cos(elem.ws)))
108      / (elem.ws * Math.cos(elem.ws) - Math.sin(elem.ws)))
109      rd.push(rdval)
110    }
111    const a = 0.409 - 0.5016 * Math.sin(elem.ws + Math.PI / 3)
112    const b = 0.6609 + 0.4767 * Math.sin(elem.ws + Math.PI / 3)
113
114    const rg = []
115    let i = 0
116    for (let h = -12; h < 12; h++) {
117      const hRad = Math.cos(deg2rad(h * 15))
118      const rgval = rd[i] * (a + b * hRad)
119      rg.push(rgval)
120      i++
121    }
122    elem.rg = rg
123    elem.rd = rd
124  })
125
126 //6. Calculo de los valores horarios de G,D y B
127 newData.meanValues.forEach((elem, index) => {
128   const hourlyValues = []
129   let hour = -12
130   const dawn = rad2deg(elem.ws) / 15
131   elem.dawn = dawn
132   for (let i = 0; i < 24; i++) {
133     const cosHour = Math.cos(deg2rad(hour * 15))
134     const cosWs = Math.cos(elem.ws)
135     if (cosHour > cosWs) {
136       const Dh = elem.rd[i] * elem.Dd0
137       const Gh = elem.rg[i] * elem.meanGR
138       const Bh = Gh - Dh

```

```

139     hourlyValues.push({
140         Bh,
141         Dh,
142         Gh,
143         cosHour,
144         cosWs,
145     })
146 } else {
147     hourlyValues.push({
148         Bh: 0,
149         Dh: 0,
150         Gh: 0,
151         cosHour,
152         cosWs,
153     })
154 }
155 hour++
156 }
157 elem.hourlyValues = hourlyValues
158 }

159
160 //8. From horizontal plane to tilted plane
161 newData.meanValues.forEach((elem, index) => {
162     for (let i = 0; i < 24; i++) {
163         const zenithRad = deg2rad(elem.zenit[i])
164         const tiltRad = deg2rad(elem.tilt[i])
165         const betaRad = deg2rad(newData.angle)
166
167         const numerator = Math.max(0, Math.cos(tiltRad))
168         const denominator = Math.cos(zenithRad)
169         const Btilt = elem.hourlyValues[i].Bh * (numerator / denominator)
170         const k1 = elem.hourlyValues[i].Bh / elem.B00[i]
171         const DcTilt = elem.hourlyValues[i].Dh * k1 * (numerator / denominator)
172         const DiTilt = elem.hourlyValues[i].Dh * (1 - k1) * ((1 + Math.cos(betaRad)) / 2)
173         const Dtilt = DcTilt + DiTilt
174         const Gtilt = Btilt + Dtilt

```

```
175     elem.hourlyValues[i] = {
176         ...elem.hourlyValues[i],
177         Btilt,
178         Dtilt,
179         DcTilt,
180         DiTilt,
181         Gtilt,
182         cosZenit: Math.cos(zenitRad),
183     }
184 }
185 })
186
187 newData.isDirtApplied = newData.applyDirtLevel == 'true'
188
189 if (newData.applyDirtLevel == 'true') {
190     newData.meanValues.forEach((elem, index) => {
191         for (let i = 0; i < 24; i++) {
192             const tiltRad = deg2rad(elem.tilt[i])
193             const betaRad = deg2rad(newData.angle)
194             let TDirtTClean = 0
195             let ar = 0
196             let c1 = 4 / (3 * Math.PI)
197             let c2 = 0
198             switch (newData.dirtLevel) {
199                 case 'CLEAN':
200                     TDirtTClean = 1
201                     ar = 0.17
202                     c2 = -0.069
203                     break
204                 case 'LOW':
205                     TDirtTClean = 0.98
206                     ar = 0.2
207                     c2 = -0.054
208                     break
209                 case 'MID':
210                     TDirtTClean = 0.97
```

```
211     ar = 0.21
212     c2 = -0.049
213     break
214
215     case 'HIGH':
216         TDirtTClean = 0.92
217         ar = 0.27
218         c2 = -0.023
219
220     default:
221         TDirtTClean = 0
222         ar = 0
223         c2 = 0
224
225     }
226
227     newData.TDirtTClean = TDirtTClean
228     newData.ar = ar
229     newData.c1 = c1
230     newData.c2 = c2
231
232     const FTB = (Math.exp(-Math.cos(tiltRad) / ar) -
233                 Math.exp(-1 / ar)) / (1 - Math.exp(-1 / ar))
234
235     const expFTD =
236
237         -(1 / ar) *
238         (c1 * (Math.sin(betaRad) + (Math.PI - betaRad)
239             - Math.sin(betaRad)) / (1 + Math.cos(betaRad))) +
240
241         c2 * Math.pow(Math.sin(betaRad) + (Math.PI -
242             betaRad - Math.sin(betaRad)) / (1 + Math.cos(betaRad)), 2))
243
244     const FTD = Math.exp(expFTD)
245
246
247     const Btilt = elem.hourlyValues[i].Btilt * TDirtTClean * (1 - FTB)
248     const DiTilt = elem.hourlyValues[i].DiTilt * TDirtTClean * (1 - FTD)
249     const DcTilt = elem.hourlyValues[i].DcTilt * TDirtTClean * (1 - FTB)
250
251     const Dtilt = DiTilt + DcTilt
252     const Gtilt = Dtilt + Btilt
253
254
255     //Overwrite data with new values
256     elem.hourlyValues[i].Btilt = Btilt
257     elem.hourlyValues[i].DiTilt = DiTilt
258     elem.hourlyValues[i].DcTilt = DcTilt
```

```

247     elem.hourlyValues[i].Dtilt = Dtilt
248     elem.hourlyValues[i].Gtilt = Gtilt
249   }
250 }
251 }
252 //9. Elección de valores clave
253 newData.meanValues.forEach((elem, index) => {
254   const significantValues = []
255   elem.hourlyValues.forEach((value, index) => {
256     if (Math.abs(value.hour) < Math.abs(elem.dawn)) {
257       significantValues.push(value)
258     }
259   })
260   elem.significantValues = significantValues
261 })
262
263 //9.1 Checking results
264 newData.meanValues.forEach((elem, index) => {
265   let totalGr = 0
266   let totalGtilt = 0
267   elem.significantValues.forEach((value, index) => {
268     totalGr += value.Gh
269     totalGtilt += value.Gtilt
270   })
271 })
272
273 return newData
274 }
275

```

Extracto de código A.2: Server/helpers/helpers.js

### A.1.3. SolarData.js

```

1 const mongoose = require('mongoose');
2 const Schema = mongoose.Schema;

```

```

3 mongoose.Promise = global.Promise;
4
5 const solarDataSchema = new Schema({
6   location: {
7     type: {
8       type: String,
9       default: 'Point'
10    },
11   coordinates: [
12     {
13       type: Number,
14       required: 'You must supply coordinates!'
15     }
16   ]
17 },
18 meanValues: [
19   {
20     type: Number
21   }
22 ]
23 });
24
25 solarDataSchema.index({ location: '2dsphere' });
26
27 module.exports = mongoose.model('SolarData', solarDataSchema);

```

Extracto de código A.3: Server/models/SolarData.js

#### A.1.4. StationsData.js

```

1 const mongoose = require('mongoose');
2 const Schema = mongoose.Schema;
3 mongoose.Promise = global.Promise;
4
5 const stationDataSchema = new Schema({
6   location: {

```

```

7   type: {
8     type: String,
9     default: 'Point'
10    },
11   coordinates: [
12     {
13       type: Number,
14       required: 'You must supply coordinates!'
15     }
16   ]
17 },
18 emaId: {
19   type: String
20 }
21 );
22
23 stationDataSchema.index({ location: '2dsphere' });
24
25 module.exports = mongoose.model('StationData', stationDataSchema);

```

Extracto de código A.4: Server/models/StationsData.js

### A.1.5. Router.js

```

1 const express = require('express');
2 const router = express.Router();
3
4 const dataController = require('../controllers/dataController');
5
6 // Do work here
7 router.get('/', (req, res) => {
8   res.send('Get your solar data from /solar-data');
9 });
10
11 router.get('/solar-data', dataController.getData);
12

```

```

13 //router.get('/save-solar-data', dataController.saveData);
14 //This route is used to save the solar data to the db
15   from the json file (Only do once)
16
17 //router.get('/save-stations', dataController.saveStations);
18 // This route is used to save the EMA ID of stations
19 in Spain by coordinates from the AEMET OpenData API
20 // (In case API fails & to not abuse their API) (Only do once)
21
22 router.get('/do-calculations', dataController.doCalculations);
23
24 router.get('/temp-profile', dataController.getTemperatureProfile);
25
26 module.exports = router;

```

Extracto de código A.5: Server/routes/index.js

### A.1.6. App.js

```

1 const express = require('express');
2 const bodyParser = require('body-parser');
3
4 const routes = require('./routes/index');
5 const app = express();
6
7 app.use(bodyParser.urlencoded({ extended: true }));
8
9 app.use(function(req, res, next) {
10   res.header('Access-Control-Allow-Origin', '*');
11   res.header('Access-Control-Allow-Headers', 'Origin,
12   X-Requested-With, Content-Type, Accept');
13   next();
14 });
15
16 app.use('/', routes);
17

```

```
18 module.exports = app;
```

Extracto de código A.6: Server/App.js

### A.1.7. Start.js

```
1 const fs = require('fs');
2 const http = require('http');
3 const https = require('https');
4 const mongoose = require('mongoose');
5
6 // Certificate
7 const privateKey = fs.readFileSync('/etc/letsencrypt/live
8 /solar-calc.ionut.cc/privkey.pem', 'utf8');
9 const certificate = fs.readFileSync('/etc/letsencrypt/live/
10 solar-calc.ionut.cc/cert.pem', 'utf8');
11 const ca = fs.readFileSync('/etc/letsencrypt/live/
12 solar-calc.ionut.cc/chain.pem', 'utf8');
13
14 const credentials = {
15   key: privateKey,
16   cert: certificate,
17   ca: ca
18 };
19
20 // Make sure we are running node 7.6+
21 const [major, minor] = process.versions.node.split('.').map(parseFloat);
22 if (major < 7 || (major === 7 && minor <= 5)) {
23   console.log('Wrong version. Update your nodejs version');
24   process.exit();
25 }
26
27 // import environmental variables from our variables.env file
28 require('dotenv').config({ path: 'variables.env' });
29
30 // Connect to our Database and handle any bad connections
```

```

31 mongoose.connect(process.env.DATABASE);
32 mongoose.Promise = global.Promise; // Tell Mongoose to use ES6 promises
33 mongoose.connection.on('error', (err) => {
34   console.error(`Error ${err.message}`);
35 });
36
37 require('./models/SolarData');
38 require('./models/StationsData');
39
40
41 const app = require('./app');
42
43 // Starting both http & https servers
44 const httpServer = http.createServer(app);
45 const httpsServer = https.createServer(credentials, app);
46
47 httpServer.listen(80, () => {
48   console.log('HTTP Server running on port 80');
49 });
50
51 httpsServer.listen(443, () => {
52   console.log('HTTPS Server running on port 443');
53 });

```

Extracto de código A.7: Server/start.js

## A.2. Código de cliente

### A.2.1. dataAquisition.js

```

1 const googleEndpoint = 'https://maps.googleapis.com/maps/api/geocode/json?';
2 const serverEndpoint = 'https://solar-calc.ionut.cc';
3
4 const addressInput = document.querySelector('#address');
5 const cityInput = document.querySelector('#city');
6 const postalInput = document.querySelector('#postal');

```

```
7 const coordBtn = document.querySelector('#button-get-coords');
8 const coordsContainer = document.querySelector('#coords-container');
9 const slope = document.querySelector('#slope');
10 const area = document.querySelector('#area');
11 const orientation = document.querySelector('#orientation');
12 const dirtLevel = document.querySelector('#dirt-level');
13 const applyDirt = document.querySelector('#apply-dirt');
14 const tableBody = document.querySelector('#radiation-data');

15
16 let calcData = {};
17 const googleApiKey = API_KEYS.GOOGLE_API_KEY;
18
19 coordBtn.addEventListener('click', getCalcData);

20
21 const moduleData = {
22   Gstar: 1000,
23   VocStar: 46.4,
24   IscStar: 9.05,
25   VmppStar: 37.4,
26   ImppStar: 8.56,
27   Ncs: 12,
28   Ncp: 6,
29   TONC: 45,
30   Tc: 25,
31   Vt: 0.025,
32   m: 1.3,
33   moduleArea: 1.957 * 0.992,
34   nominalPower: 320
35 };
36
37 const generatordata = {
38   seriesModules: 12,
39   parallelModules: 11
40 };
41
42 const inverterData = {
```

```
43     k0: 0.01,
44     k1: 0.025,
45     k2: 0.05,
46     power: 40000,
47     vMin: 420,
48     vMax: 750
49   };
50
51   async function getCalcData() {
52     coordBtn.textContent = 'Calculando';
53     coordBtn.disabled = true;
54     coordBtn.classList.add('is-loading');
55
56     calcData.placement = await getCoordinates();
57
58     calcData.surfaceInfo = {
59       area: area.value,
60       slope: slope.value,
61       orientation: orientation.value,
62       dirtLevel: dirtLevel.value,
63       applyDirt: true
64     };
65     const radiationData = await doCalculations(calcData);
66     const cellTempProfile = await calculateCellTemp(calcData, radiationData);
67     console.log(cellTempProfile);
68     const VocProfile = calculateVoc(cellTempProfile);
69     const IscProfile = calculateIsc(radiationData);
70     const { VmppProfile, ImppProfile, RsStar, rs, koc, DMO, DM, impp, vmpp } =
71     applyVariableFF(VocProfile, IscProfile, cellTempProfile);
72     const PmppProfile = calculatePmpp(ImppProfile, VmppProfile);
73     const PdcProfile = calculateGeneratorPower(PmppProfile);
74     const PiProfile = calculatePiProfile(PdcProfile);
75     const PoProfile = calculatePoProfile(PiProfile);
76     const PacProfile = calculatePacProfile(PoProfile);
77     const MonthlyEac = calculateMonthlyEac(PacProfile);
78     const MonthlyEdc = calculateMonthlyEdc(PdcProfile);
```

```
79  const monthlyYf = MonthlyEac.map(  
80    (val) => val / (moduleData.VmppStar * moduleData.ImppStar *  
81      generatordata.seriesModules * generatordata.parallelModules)  
82  );  
83  const MonthlyEnergykWh = MonthlyEac.map((val) => val / 1000);  
84  const AnualEnergyW = calculateAnualEnergy(MonthlyEac);  
85  const AnualEnergykWh = AnualEnergyW / 1000;  
86  const anualYf = AnualEnergyW / (moduleData.VmppStar * moduleData.ImppStar *  
87    generatordata.seriesModules * generatordata.parallelModules);  
88  
89 // createTable(radiationData, MonthlyEac, MonthlyEdc, Yf, tableBody);  
90  
91 console.log({  
92   radiationData,  
93   PdcProfile,  
94   PacProfile,  
95   MppValues: { VmppProfile, ImppProfile },  
96   VocProfile,  
97   IscProfile,  
98   cellTempProfile,  
99   IscProfile,  
100  variableFF: { RsStar, rs, koc, DMO, DM, impp, vmpp }  
101});  
102  
103 saveData(  
104   radiationData,  
105   MonthlyEnergykWh,  
106   AnualEnergykWh,  
107   monthlyYf,  
108   PacProfile,  
109   anualYf,  
110   calcData.placement,  
111   calcData.surfaceInfo.area,  
112   calcData.surfaceInfo.slope,  
113   calcData.surfaceInfo.orientation  
114 );
```

```
115
116     window.location = 'results.html';
117 }
118
119 const getCoordinates = async () => {
120     const address = addressInput.value.split(' ').join('+');
121     const city = cityInput.value;
122     const postal = postalInput.value;
123     const requestURL = `${googleEndpoint}address=$
124     ${address},${city},${postal},spain&key=${googleApiKey}`;
125     const response = await fetch(requestURL);
126     const data = await response.json();
127     console.log(data);
128     const info = {
129         formattedAddress: data.results[0].formatted_address,
130         lat: data.results[0].geometry.location.lat,
131         long: data.results[0].geometry.location.lng
132     };
133
134     return info;
135 };
136
137 const doCalculations = async (calcData) => {
138     const requestURL = `${serverEndpoint}/do-calculations?latitude=
139     ${calcData.placement.lat}&longitude=${calcData.placement.long}&angle=${
140         calcData.surfaceInfo.slope
141     }&area=${calcData.surfaceInfo.area}&orientation=
142     ${calcData.surfaceInfo.orientation}&applyDirtLevel=${
143         calcData.surfaceInfo.applyDirt
144     }&dirtLevel=${calcData.surfaceInfo.dirtLevel}`;
145
146     const res = await fetch(requestURL);
147     const data = await res.json();
148     return data.data;
149 };
150
```

```
151 const calculateCellTemp = async (calcData, radiationData) => {
152   const requestURL = `${serverEndpoint}/temp-profile?latitude=
153 ${calcData.placement.lat}&longitude=${calcData.placement.long}`;
154   const res = await fetch(requestURL);
155   const data = await res.json();
156   const tempProfiles = data.profiles;
157   const cellTempProfiles = tempProfiles.map(({ hourlyTa, Tmax, Tmin }, 
158   index) => {
159     const meanValue = radiationData.meanValues[index];
160     const TCPProfile = hourlyTa.map((temp, index) => {
161       const Gef = meanValue.hourlyValues[index].Gtilt * 1000;
162       const Tc = 25 + (Gef * (moduleData.TONC - 20)) / 800;
163       //Cambiar 1000 por la G en el plano inclinado a esa hora
164       return Tc;
165     });
166     return { TCPProfile, month: index + 1, Tmax, Tmin };
167   });
168   return cellTempProfiles;
169 };
170
171 const calculateVoc = (cellTempProfile) => {
172   const VocProfile = cellTempProfile.map(({ TCPProfile }, index) => {
173     const VocArray = TCPProfile.map((temp) => {
174       const Voc = moduleData.VocStar + (temp - moduleData.Tc) *
175         (-2.3 / 1000) * moduleData.Ncs;
176       return Voc;
177     });
178     return VocArray;
179   });
180   return VocProfile;
181 };
182
183 const calculateIsc = (radiationData) => {
184   const IscProfile = radiationData.meanValues.map(({ hourlyValues }) => {
185     const IscArray = hourlyValues.map(({ Gtilt }) => {
186       const IscValue = Gtilt * 1000 *
```

```
187     (moduleData.IscStar / moduleData.Gstar);
188
189     return IscValue;
190 );
191
192 return IscArray;
193 );
194
195 const applyVariableFF = (VocProfile, IscProfile, cellTempProfile) => {
196
196   const Vtn = (moduleData.Vt * (273 + 25)) / 300;
197
198   const RsStar =
199     (moduleData.VocStar / moduleData.Ncs -
200      moduleData.VmppStar / moduleData.Ncs +
201      moduleData.m * Vtn * Math.log(1 - moduleData.ImppStar /
202        moduleData.IscStar)) /
203      (moduleData.ImppStar / moduleData.Ncp);
204
205
206   const rs = VocProfile.map((VocArray, arrIndex) => {
207
207     return VocArray.map((VocValue, valIndex) => {
208
209       return RsStar * ((moduleData.Ncs / moduleData.Ncp) *
210         (IscProfile[arrIndex][valIndex] / VocValue));
211
212     });
213
214   const koc = VocProfile.map((VocArray, arrIndex) => {
215
216     return VocArray.map((VocValue, valIndex) => {
217
218       return VocValue / moduleData.Ncs / (moduleData.m *
219         ((moduleData.Vt * (cellTempProfile[arrIndex].TCPProfile[valIndex]
220           + 273)) / 300));
221
222     });
223
224   const DMO = koc.map((kocArray, arrIndex) => {
225
226     return kocArray.map((kocValue, valIndex) => {
227
228       return (kocValue - 1) / (kocValue - Math.log(kocValue));
229
230     });
231
232   const DM = DMO.map((DMOArray, arrIndex) => {
```

```
223     return DMOArray.map((DMOValue, valIndex) => {
224         return DMOValue + 2 * rs[arrIndex][valIndex] * Math.pow(DMOValue, 2);
225     });
226 };
227
228 const impp = DM.map((DMArry, arrIndex) => {
229     return DMArry.map((DMValue, valIndex) => {
230         return 1 - DMValue / koc[arrIndex][valIndex];
231     });
232 });
233
234 const vmpp = impp.map((imppArray, arrIndex) => {
235     return imppArray.map((imppValue, valIndex) => {
236         return 1 - Math.log(koc[arrIndex][valIndex] /
237             DM[arrIndex][valIndex]) / koc[arrIndex][valIndex] -
238             rs[arrIndex][valIndex] * imppValue;
239     });
240 });
241
242 const Vmpp = vmpp.map((vmppArray, arrIndex) => {
243     return vmppArray.map((vmppValue, valIndex) => {
244         return vmppValue * VocProfile[arrIndex][valIndex];
245     });
246 });
247 const Impp = impp.map((imppArray, arrIndex) => {
248     return imppArray.map((imppValue, valIndex) => {
249         return imppValue * IscProfile[arrIndex][valIndex];
250     });
251 });
252
253 return { VmppProfile: Vmpp, ImppProfile: Impp, RsStar, rs,
254         koc, DMO, DM, impp, vmpp };
255 };
256
257 const calculatePmpp = (ImppProfile, VmppProfile) => {
258     const PmppProfile = ImppProfile.map((ImppArray, dayIndex) => {
```

```
259     const PmppArray = ImppArray.map((ImppValue, hourIndex) => {
260         return ImppValue * VmppProfile[dayIndex][hourIndex];
261     });
262     return PmppArray;
263 );
264 return PmppProfile;
265 };
266
267 const calculateGeneratorPower = (PmppProfile) => {
268     const PdcProfile = PmppProfile.map((PmppArray) => {
269         return PmppArray.map((PmppValue) => {
270             return PmppValue * generatordata.seriesModules * generatordata.parallelModules;
271         });
272     });
273     return PdcProfile;
274 };
275
276 const calculatePiProfile = (PdcProfile) => {
277     return PdcProfile.map((PdcArray) => {
278         return PdcArray.map((PdcValue) => {
279             return PdcValue / inverterData.power;
280         });
281     });
282 };
283
284 const calculatePoProfile = (PiProfile) => {
285     return PiProfile.map((PiArray) => {
286         return PiArray.map((PiValue) => {
287             if (PiValue <= 0) {
288                 return 0;
289             }
290             const a = inverterData.k2;
291             const b = inverterData.k1 + 1;
292             const c = inverterData.k0 - PiValue;
293             const firstSolution = (-b + Math.sqrt(b * b - 4 * a * c)) / (2 * a);
294             const secondSolution = (-b - Math.sqrt(b * b - 4 * a * c)) / (2 * a);
```

```
295     return firstSolution > 0 ? firstSolution : secondSolution;
296   });
297 });
298 };
299
300 const calculatePacProfile = (PoProfile) => {
301   return PoProfile.map((PoArray) => {
302     return PoArray.map((PoValue) => {
303       return PoValue * inverterData.power;
304     });
305   });
306 };
307
308 const calculateMonthlyEac = (PacProfile) => {
309   return PacProfile.map((PacArray) => {
310     return (
311       30 *
312       PacArray.reduce((total, currentValue) => {
313         return total + (currentValue > 0 ? currentValue : 0);
314         EacCell.textContent = MonthlyEac[index];
315       }, 0)
316     );
317   });
318 };
319
320 const calculateMonthlyEdc = (PdcProfile) => {
321   return PdcProfile.map((PdcArray) => {
322     return (
323       30 *
324       PdcArray.reduce((total, currentValue) => {
325         return total + (currentValue > 0 ? currentValue : 0);
326       }, 0)
327     );
328   });
329 };
330 }
```

```
331 const calculateAnualEnergy = (MonthlyEnergy) => {
332   return MonthlyEnergy.reduce((total, currentValue) => {
333     return total + (currentValue > 0 ? currentValue : 0);
334   }, 0);
335 };
336
337 const createTable = (radiationData, MonthlyEac, MonthlyEdc, Yf, tableBody) => {
338   tableBody.innerHTML = '';
339   const { meanValues } = radiationData;
340
341   meanValues.forEach((value, index) => {
342     const row = document.createElement('tr');
343     let Gefd = 0;
344     let Defd = 0;
345     let Befd = 0;
346
347     value.hourlyValues.forEach((hValue) => {
348       Gefd += hValue.Gtilt;
349       Defd += hValue.Dtilt;
350       Befd += hValue.Btilt;
351     });
352     const G0dCell = document.createElement('td');
353     G0dCell.textContent = value.meanGR;
354     const Bd0Cell = document.createElement('td');
355     Bd0Cell.textContent = value.Dd0;
356     const Dd0dCell = document.createElement('td');
357     Dd0dCell.textContent = value.Bd0;
358     const GefdCell = document.createElement('td');
359     GefdCell.textContent = Gefd;
360     const DefdCell = document.createElement('td');
361     DefdCell.textContent = Defd;
362     const BefdCell = document.createElement('td');
363     BefdCell.textContent = Befd;
364     const monthCell = document.createElement('td');
365     monthCell.textContent = index + 1;
366     const EacCell = document.createElement('td');
```

```
367     EacCell.textContent = MonthlyEac[index] / 1000;
368     const EdcCell = document.createElement('td');
369     EdcCell.textContent = MonthlyEdc[index] / 1000;
370     const YfCell = document.createElement('td');
371     YfCell.textContent = Yf[index];
372     row.appendChild(monthCell);
373     row.appendChild(G0dCell);
374     row.appendChild(Bd0Cell);
375     row.appendChild(Dd0dCell);
376     row.appendChild(GefdCell);
377     row.appendChild(DefdCell);
378     row.appendChild(BeFdCell);
379     row.appendChild(EacCell);
380     row.appendChild(EdcCell);
381     row.appendChild(YfCell);
382     tableBody.appendChild(row);
383   });
384 };
385
386 const saveData = (radiationData, MonthlyEnergykWh, AnualEnergykWh,
387 monthlyYf, PacProfile, AnualYf, location,
388 area, slope, orientation) => {
389   const totalModules = generatordata.seriesModules * generatordata.parallelModules;
390   const totalArea = moduleData.moduleArea * totalModules;
391   const areaRelation = area / totalArea;
392   const installedModules = parseInt(areaRelation * totalModules);
393   const installedPower = installedModules * moduleData.nominalPower;
394   const inverterPower = installedPower * 0.9;
395
396   const results = {
397     placementData: {
398       area,
399       location,
400       slope,
401       orientation
402     },
403     radiationData,
404     monthlyYf,
405     monthlyEnergykWh,
406     anualYf,
407     anualEnergykWh,
408     inverterPower,
409     installedPower,
410     totalArea,
411     totalModules,
412     areaRelation,
413     installedModules,
414     moduleData,
415     generatordata,
416     pacProfile,
417     pacProfile,
418     generatordata,
419     radiationData,
420     monthlyEnergykWh,
421     anualEnergykWh,
422     monthlyYf,
423     anualYf,
424     inverterPower,
425     installedPower,
426     totalArea,
427     totalModules,
428     areaRelation,
429     installedModules,
430     moduleData,
431     generatordata,
432     pacProfile,
433     pacProfile,
434     generatordata,
435     radiationData,
436     monthlyEnergykWh,
437     anualEnergykWh,
438     monthlyYf,
439     anualYf,
440     inverterPower,
441     installedPower,
442     totalArea,
443     totalModules,
444     areaRelation,
445     installedModules,
446     moduleData,
447     generatordata,
448     pacProfile,
449     pacProfile,
450     generatordata,
451     radiationData,
452     monthlyEnergykWh,
453     anualEnergykWh,
454     monthlyYf,
455     anualYf,
456     inverterPower,
457     installedPower,
458     totalArea,
459     totalModules,
460     areaRelation,
461     installedModules,
462     moduleData,
463     generatordata,
464     pacProfile,
465     pacProfile,
466     generatordata,
467     radiationData,
468     monthlyEnergykWh,
469     anualEnergykWh,
470     monthlyYf,
471     anualYf,
472     inverterPower,
473     installedPower,
474     totalArea,
475     totalModules,
476     areaRelation,
477     installedModules,
478     moduleData,
479     generatordata,
480     pacProfile,
481     pacProfile,
482     generatordata,
483     radiationData,
484     monthlyEnergykWh,
485     anualEnergykWh,
486     monthlyYf,
487     anualYf,
488     inverterPower,
489     installedPower,
490     totalArea,
491     totalModules,
492     areaRelation,
493     installedModules,
494     moduleData,
495     generatordata,
496     pacProfile,
497     pacProfile,
498     generatordata,
499     radiationData,
500     monthlyEnergykWh,
501     anualEnergykWh,
502     monthlyYf,
503     anualYf,
504     inverterPower,
505     installedPower,
506     totalArea,
507     totalModules,
508     areaRelation,
509     installedModules,
510     moduleData,
511     generatordata,
512     pacProfile,
513     pacProfile,
514     generatordata,
515     radiationData,
516     monthlyEnergykWh,
517     anualEnergykWh,
518     monthlyYf,
519     anualYf,
520     inverterPower,
521     installedPower,
522     totalArea,
523     totalModules,
524     areaRelation,
525     installedModules,
526     moduleData,
527     generatordata,
528     pacProfile,
529     pacProfile,
530     generatordata,
531     radiationData,
532     monthlyEnergykWh,
533     anualEnergykWh,
534     monthlyYf,
535     anualYf,
536     inverterPower,
537     installedPower,
538     totalArea,
539     totalModules,
540     areaRelation,
541     installedModules,
542     moduleData,
543     generatordata,
544     pacProfile,
545     pacProfile,
546     generatordata,
547     radiationData,
548     monthlyEnergykWh,
549     anualEnergykWh,
550     monthlyYf,
551     anualYf,
552     inverterPower,
553     installedPower,
554     totalArea,
555     totalModules,
556     areaRelation,
557     installedModules,
558     moduleData,
559     generatordata,
560     pacProfile,
561     pacProfile,
562     generatordata,
563     radiationData,
564     monthlyEnergykWh,
565     anualEnergykWh,
566     monthlyYf,
567     anualYf,
568     inverterPower,
569     installedPower,
570     totalArea,
571     totalModules,
572     areaRelation,
573     installedModules,
574     moduleData,
575     generatordata,
576     pacProfile,
577     pacProfile,
578     generatordata,
579     radiationData,
580     monthlyEnergykWh,
581     anualEnergykWh,
582     monthlyYf,
583     anualYf,
584     inverterPower,
585     installedPower,
586     totalArea,
587     totalModules,
588     areaRelation,
589     installedModules,
590     moduleData,
591     generatordata,
592     pacProfile,
593     pacProfile,
594     generatordata,
595     radiationData,
596     monthlyEnergykWh,
597     anualEnergykWh,
598     monthlyYf,
599     anualYf,
600     inverterPower,
601     installedPower,
602     totalArea,
603     totalModules,
604     areaRelation,
605     installedModules,
606     moduleData,
607     generatordata,
608     pacProfile,
609     pacProfile,
610     generatordata,
611     radiationData,
612     monthlyEnergykWh,
613     anualEnergykWh,
614     monthlyYf,
615     anualYf,
616     inverterPower,
617     installedPower,
618     totalArea,
619     totalModules,
620     areaRelation,
621     installedModules,
622     moduleData,
623     generatordata,
624     pacProfile,
625     pacProfile,
626     generatordata,
627     radiationData,
628     monthlyEnergykWh,
629     anualEnergykWh,
630     monthlyYf,
631     anualYf,
632     inverterPower,
633     installedPower,
634     totalArea,
635     totalModules,
636     areaRelation,
637     installedModules,
638     moduleData,
639     generatordata,
640     pacProfile,
641     pacProfile,
642     generatordata,
643     radiationData,
644     monthlyEnergykWh,
645     anualEnergykWh,
646     monthlyYf,
647     anualYf,
648     inverterPower,
649     installedPower,
650     totalArea,
651     totalModules,
652     areaRelation,
653     installedModules,
654     moduleData,
655     generatordata,
656     pacProfile,
657     pacProfile,
658     generatordata,
659     radiationData,
660     monthlyEnergykWh,
661     anualEnergykWh,
662     monthlyYf,
663     anualYf,
664     inverterPower,
665     installedPower,
666     totalArea,
667     totalModules,
668     areaRelation,
669     installedModules,
670     moduleData,
671     generatordata,
672     pacProfile,
673     pacProfile,
674     generatordata,
675     radiationData,
676     monthlyEnergykWh,
677     anualEnergykWh,
678     monthlyYf,
679     anualYf,
680     inverterPower,
681     installedPower,
682     totalArea,
683     totalModules,
684     areaRelation,
685     installedModules,
686     moduleData,
687     generatordata,
688     pacProfile,
689     pacProfile,
690     generatordata,
691     radiationData,
692     monthlyEnergykWh,
693     anualEnergykWh,
694     monthlyYf,
695     anualYf,
696     inverterPower,
697     installedPower,
698     totalArea,
699     totalModules,
700     areaRelation,
701     installedModules,
702     moduleData,
703     generatordata,
704     pacProfile,
705     pacProfile,
706     generatordata,
707     radiationData,
708     monthlyEnergykWh,
709     anualEnergykWh,
710     monthlyYf,
711     anualYf,
712     inverterPower,
713     installedPower,
714     totalArea,
715     totalModules,
716     areaRelation,
717     installedModules,
718     moduleData,
719     generatordata,
720     pacProfile,
721     pacProfile,
722     generatordata,
723     radiationData,
724     monthlyEnergykWh,
725     anualEnergykWh,
726     monthlyYf,
727     anualYf,
728     inverterPower,
729     installedPower,
730     totalArea,
731     totalModules,
732     areaRelation,
733     installedModules,
734     moduleData,
735     generatordata,
736     pacProfile,
737     pacProfile,
738     generatordata,
739     radiationData,
740     monthlyEnergykWh,
741     anualEnergykWh,
742     monthlyYf,
743     anualYf,
744     inverterPower,
745     installedPower,
746     totalArea,
747     totalModules,
748     areaRelation,
749     installedModules,
750     moduleData,
751     generatordata,
752     pacProfile,
753     pacProfile,
754     generatordata,
755     radiationData,
756     monthlyEnergykWh,
757     anualEnergykWh,
758     monthlyYf,
759     anualYf,
760     inverterPower,
761     installedPower,
762     totalArea,
763     totalModules,
764     areaRelation,
765     installedModules,
766     moduleData,
767     generatordata,
768     pacProfile,
769     pacProfile,
770     generatordata,
771     radiationData,
772     monthlyEnergykWh,
773     anualEnergykWh,
774     monthlyYf,
775     anualYf,
776     inverterPower,
777     installedPower,
778     totalArea,
779     totalModules,
780     areaRelation,
781     installedModules,
782     moduleData,
783     generatordata,
784     pacProfile,
785     pacProfile,
786     generatordata,
787     radiationData,
788     monthlyEnergykWh,
789     anualEnergykWh,
790     monthlyYf,
791     anualYf,
792     inverterPower,
793     installedPower,
794     totalArea,
795     totalModules,
796     areaRelation,
797     installedModules,
798     moduleData,
799     generatordata,
800     pacProfile,
801     pacProfile,
802     generatordata,
803     radiationData,
804     monthlyEnergykWh,
805     anualEnergykWh,
806     monthlyYf,
807     anualYf,
808     inverterPower,
809     installedPower,
810     totalArea,
811     totalModules,
812     areaRelation,
813     installedModules,
814     moduleData,
815     generatordata,
816     pacProfile,
817     pacProfile,
818     generatordata,
819     radiationData,
820     monthlyEnergykWh,
821     anualEnergykWh,
822     monthlyYf,
823     anualYf,
824     inverterPower,
825     installedPower,
826     totalArea,
827     totalModules,
828     areaRelation,
829     installedModules,
830     moduleData,
831     generatordata,
832     pacProfile,
833     pacProfile,
834     generatordata,
835     radiationData,
836     monthlyEnergykWh,
837     anualEnergykWh,
838     monthlyYf,
839     anualYf,
840     inverterPower,
841     installedPower,
842     totalArea,
843     totalModules,
844     areaRelation,
845     installedModules,
846     moduleData,
847     generatordata,
848     pacProfile,
849     pacProfile,
850     generatordata,
851     radiationData,
852     monthlyEnergykWh,
853     anualEnergykWh,
854     monthlyYf,
855     anualYf,
856     inverterPower,
857     installedPower,
858     totalArea,
859     totalModules,
860     areaRelation,
861     installedModules,
862     moduleData,
863     generatordata,
864     pacProfile,
865     pacProfile,
866     generatordata,
867     radiationData,
868     monthlyEnergykWh,
869     anualEnergykWh,
870     monthlyYf,
871     anualYf,
872     inverterPower,
873     installedPower,
874     totalArea,
875     totalModules,
876     areaRelation,
877     installedModules,
878     moduleData,
879     generatordata,
880     pacProfile,
881     pacProfile,
882     generatordata,
883     radiationData,
884     monthlyEnergykWh,
885     anualEnergykWh,
886     monthlyYf,
887     anualYf,
888     inverterPower,
889     installedPower,
890     totalArea,
891     totalModules,
892     areaRelation,
893     installedModules,
894     moduleData,
895     generatordata,
896     pacProfile,
897     pacProfile,
898     generatordata,
899     radiationData,
900     monthlyEnergykWh,
901     anualEnergykWh,
902     monthlyYf,
903     anualYf,
904     inverterPower,
905     installedPower,
906     totalArea,
907     totalModules,
908     areaRelation,
909     installedModules,
910     moduleData,
911     generatordata,
912     pacProfile,
913     pacProfile,
914     generatordata,
915     radiationData,
916     monthlyEnergykWh,
917     anualEnergykWh,
918     monthlyYf,
919     anualYf,
920     inverterPower,
921     installedPower,
922     totalArea,
923     totalModules,
924     areaRelation,
925     installedModules,
926     moduleData,
927     generatordata,
928     pacProfile,
929     pacProfile,
930     generatordata,
931     radiationData,
932     monthlyEnergykWh,
933     anualEnergykWh,
934     monthlyYf,
935     anualYf,
936     inverterPower,
937     installedPower,
938     totalArea,
939     totalModules,
940     areaRelation,
941     installedModules,
942     moduleData,
943     generatordata,
944     pacProfile,
945     pacProfile,
946     generatordata,
947     radiationData,
948     monthlyEnergykWh,
949     anualEnergykWh,
950     monthlyYf,
951     anualYf,
952     inverterPower,
953     installedPower,
954     totalArea,
955     totalModules,
956     areaRelation,
957     installedModules,
958     moduleData,
959     generatordata,
960     pacProfile,
961     pacProfile,
962     generatordata,
963     radiationData,
964     monthlyEnergykWh,
965     anualEnergykWh,
966     monthlyYf,
967     anualYf,
968     inverterPower,
969     installedPower,
970     totalArea,
971     totalModules,
972     areaRelation,
973     installedModules,
974     moduleData,
975     generatordata,
976     pacProfile,
977     pacProfile,
978     generatordata,
979     radiationData,
980     monthlyEnergykWh,
981     anualEnergykWh,
982     monthlyYf,
983     anualYf,
984     inverterPower,
985     installedPower,
986     totalArea,
987     totalModules,
988     areaRelation,
989     installedModules,
990     moduleData,
991     generatordata,
992     pacProfile,
993     pacProfile,
994     generatordata,
995     radiationData,
996     monthlyEnergykWh,
997     anualEnergykWh,
998     monthlyYf,
999     anualYf,
1000    inverterPower,
1001    installedPower,
1002    totalArea,
1003    totalModules,
1004    areaRelation,
1005    installedModules,
1006    moduleData,
1007    generatordata,
1008    pacProfile,
1009    pacProfile,
1010    generatordata,
1011    radiationData,
1012    monthlyEnergykWh,
1013    anualEnergykWh,
1014    monthlyYf,
1015    anualYf,
1016    inverterPower,
1017    installedPower,
1018    totalArea,
1019    totalModules,
1020    areaRelation,
1021    installedModules,
1022    moduleData,
1023    generatordata,
1024    pacProfile,
1025    pacProfile,
1026    generatordata,
1027    radiationData,
1028    monthlyEnergykWh,
1029    anualEnergykWh,
1030    monthlyYf,
1031    anualYf,
1032    inverterPower,
1033    installedPower,
1034    totalArea,
1035    totalModules,
1036    areaRelation,
1037    installedModules,
1038    moduleData,
1039    generatordata,
1040    pacProfile,
1041    pacProfile,
1042    generatordata,
1043    radiationData,
1044    monthlyEnergykWh,
1045    anualEnergykWh,
1046    monthlyYf,
1047    anualYf,
1048    inverterPower,
1049    installedPower,
1050    totalArea,
1051    totalModules,
1052    areaRelation,
1053    installedModules,
1054    moduleData,
1055    generatordata,
1056    pacProfile,
1057    pacProfile,
1058    generatordata,
1059    radiationData,
1060    monthlyEnergykWh,
1061    anualEnergykWh,
1062    monthlyYf,
1063    anualYf,
1064    inverterPower,
1065    installedPower,
1066    totalArea,
1067    totalModules,
1068    areaRelation,
1069    installedModules,
1070    moduleData,
1071    generatordata,
1072    pacProfile,
1073    pacProfile,
1074    generatordata,
1075    radiationData,
1076    monthlyEnergykWh,
1077    anualEnergykWh,
1078    monthlyYf,
1079    anualYf,
1080    inverterPower,
1081    installedPower,
1082    totalArea,
1083    totalModules,
1084    areaRelation,
1085    installedModules,
1086    moduleData,
1087    generatordata,
1088    pacProfile,
1089    pacProfile,
1090    generatordata,
1091    radiationData,
1092    monthlyEnergykWh,
1093    anualEnergykWh,
1094    monthlyYf,
1095    anualYf,
1096    inverterPower,
1097    installedPower,
1098    totalArea,
1099    totalModules,
1100    areaRelation,
1101    installedModules,
1102    moduleData,
1103    generatordata,
1104    pacProfile,
1105    pacProfile,
1106    generatordata,
1107    radiationData,
1108    monthlyEnergykWh,
1109    anualEnergykWh,
1110    monthlyYf,
1111    anualYf,
1112    inverterPower,
1113    installedPower,
1114    totalArea,
1115    totalModules,
1116    areaRelation,
1117    installedModules,
1118    moduleData,
1119    generatordata,
1120    pacProfile,
1121    pacProfile,
1122    generatordata,
1123    radiationData,
1124    monthlyEnergykWh,
1125    anualEnergykWh,
1126    monthlyYf,
1127    anualYf,
1128    inverterPower,
1129    installedPower,
1130    totalArea,
1131    totalModules,
1132    areaRelation,
1133    installedModules,
1134    moduleData,
1135    generatordata,
1136    pacProfile,
1137    pacProfile,
1138    generatordata,
1139    radiationData,
1140    monthlyEnergykWh,
1141    anualEnergykWh,
1142    monthlyYf,
1143    anualYf,
1144    inverterPower,
1145    installedPower,
1146    totalArea,
1147    totalModules,
1148    areaRelation,
1149    installedModules,
1150    moduleData,
1151    generatordata,
1152    pacProfile,
1153    pacProfile,
1154    generatordata,
1155    radiationData,
1156    monthlyEnergykWh,
1157    anualEnergykWh,
1158    monthlyYf,
1159    anualYf,
1160    inverterPower,
1161    installedPower,
1162    totalArea,
1163    totalModules,
1164    areaRelation,
1165    installedModules,
1166    moduleData,
1167    generatordata,
1168    pacProfile,
1169    pacProfile,
1170    generatordata,
1171    radiationData,
1172    monthlyEnergykWh,
1173    anualEnergykWh,
1174    monthlyYf,
1175    anualYf,
1176    inverterPower,
1177    installedPower,
1178    totalArea,
1179    totalModules,
1180    areaRelation,
1181    installedModules,
1182    moduleData,
1183    generatordata,
1184    pacProfile,
1185    pacProfile,
1186    generatordata,
1187    radiationData,
1188    monthlyEnergykWh,
1189    anualEnergykWh,
1190    monthlyYf,
1191    anualYf,
1192    inverterPower,
1193    installedPower,
1194    totalArea,
1195    totalModules,
1196    areaRelation,
1197    installedModules,
1198    moduleData,
1199    generatordata,
1200    pacProfile,
1201    pacProfile,
1202    generatordata,
1203    radiationData,
1204    monthlyEnergykWh,
1205    anualEnergykWh,
1206    monthlyYf,
1207    anualYf,
1208    inverterPower,
1209    installedPower,
1210    totalArea,
1211    totalModules,
1212    areaRelation,
1213    installedModules,
1214    moduleData,
1215    generatordata,
1216    pacProfile,
1217    pacProfile,
1218    generatordata,
1219    radiationData,
1220    monthlyEnergykWh,
1221    anualEnergykWh,
1222    monthlyYf,
1223    anualYf,
1224    inverterPower,
1225    installedPower,
1226    totalArea,
1227    totalModules,
1228    areaRelation,
1229    installedModules,
1230    moduleData,
1231    generatordata,
1232    pacProfile,
1233    pacProfile,
1234    generatordata,
1235    radiationData,
1236    monthlyEnergykWh,
1237    anualEnergykWh,
1238    monthlyYf,
1239    anualYf,
1240    inverterPower,
1241    installedPower,
1242    totalArea,
1243    totalModules,
1244    areaRelation,
1245    installedModules,
1246    moduleData,
1247    generatordata,
1248    pacProfile,
1249    pacProfile,
1250    generatordata,
1251    radiationData,
1252    monthlyEnergykWh,
1253    anualEnergykWh,
1254    monthlyYf,
1255    anualYf,
1256    inverterPower,
1257    installedPower,
1258    totalArea,
1259    totalModules,
1260    areaRelation,
1261    installedModules,
1262    moduleData,
1263    generatordata,
1264    pacProfile,
1265    pacProfile,
1266    generatordata,
1267    radiationData,
1268    monthlyEnergykWh,
1269    anualEnergykWh,
1270    monthlyYf,
1271    anualYf,
1272    inverterPower,
1273    installedPower,
1274    totalArea,
1275    totalModules,
1276    areaRelation,
1277    installedModules,
1278    moduleData,
1279    generatordata,
1280    pacProfile,
1281    pacProfile,
1282    generatordata,
1283    radiationData,
1284    monthlyEnergykWh,
1285    anualEnergykWh,
1286    monthlyYf,
1287    anualYf,
1288    inverterPower,
1289    installedPower,
1290    totalArea,
1291    totalModules,
1292    areaRelation,
1293    installedModules,
1294    moduleData,
1295    generatordata,
1296    pacProfile,
1297    pacProfile,
1298    generatordata,
1299    radiationData,
1300    monthlyEnergykWh,
1301    anualEnergykWh,
1302    monthlyYf,
1303    anualYf,
1304    inverterPower,
1305    installedPower,
1306    totalArea,
1307    totalModules,
1308    areaRelation,
1309    installedModules,
1310    moduleData,
1311    generatordata,
1312    pacProfile,
1313    pacProfile,
1314    generatordata,
1315    radiationData,
1316    monthlyEnergykWh,
1317    anualEnergykWh,
1318    monthlyYf,
1319    anualYf,
1320    inverterPower,
1321    installedPower,
1322    totalArea,
1323    totalModules,
1324    areaRelation,
1325    installedModules,
1326    moduleData,
1327    generatordata,
1328    pacProfile,
1329    pacProfile,
1330    generatordata,
1331    radiationData,
1332    monthlyEnergykWh,
1333    anualEnergykWh,
1334    monthlyYf,
1335    anualYf,
1336    inverterPower,
1337    installedPower,
1338    totalArea,
1339    totalModules,
1340    areaRelation,
1341    installedModules,
1342    moduleData,
1343    generatordata,
1344    pacProfile,
1345    pacProfile,
1346    generatordata,
1347    radiationData,
1348    monthlyEnergykWh,
1349    anualEnergykWh,
1350    monthlyYf,
1351    anualYf,
1352    inverterPower,
1353    installedPower,
1354    totalArea,
1355    totalModules,
1356    areaRelation,
1357    installedModules,
1358    moduleData,
1359    generatordata,
1360    pacProfile,
1361    pacProfile,
1362    generatordata,
1363    radiationData,
1364    monthlyEnergykWh,
1365    anualEnergykWh,
1366    monthlyYf,
1367    anualYf,
1368    inverterPower,
1369    installedPower,
1370    totalArea,
1371    totalModules,
1372    areaRelation,
1373    installedModules,
1374    moduleData,
1375    generatordata,
1376    pacProfile,
1377    pacProfile,
1378    generatordata,
1379    radiationData,
1380    monthlyEnergykWh,
1381    anualEnergykWh,
1382    monthlyYf,
1383    anualYf,
1384    inverterPower,
1385    installedPower,
1386    totalArea,
1387    totalModules,
1388    areaRelation,
1389    installedModules,
1390    moduleData,
1391    generatordata,
1392    pacProfile,
1393    pacProfile,
1394    generatordata,
1395    radiationData,
1396    monthlyEnergykWh,
1397    anualEnergykWh,
1398    monthlyYf,
1399    anualYf,
1400    inverterPower,
1401    installedPower,
1402    totalArea,
1403    totalModules,
1404    areaRelation,
1405    installedModules,
1406    moduleData,
1407    generatordata,
1408    pacProfile,
1409    pacProfile,
1410    generatordata,
1411    radiationData,
1412    monthlyEnergykWh,
1413    anualEnergykWh,
1414    monthlyYf,
1415    anualYf,
1416    inverterPower,
1417    installedPower,
1418    totalArea,
1419    totalModules,
1420    areaRelation,
1421    installedModules,
1422    moduleData,
1423    generatordata,
1424    pacProfile,
1425    pacProfile,
1426    generatordata,
1427    radiationData,
1428    monthlyEnergykWh,
1429    anualEnergykWh,
1430    monthlyYf,
1431    anualYf,
1432    inverterPower,
1433    installedPower,
1434    totalArea,
1435    totalModules,
1436    areaRelation,
1437    installedModules,
1438    moduleData,
1439    generatordata,
1440    pacProfile,
1441    pacProfile,
1442    generatordata,
1443    radiationData,
1444    monthlyEnergykWh,
1445    anualEnergykWh,
1446    monthlyYf,
1447    anualYf,
1448    inverterPower,
1449    installedPower,
1450    totalArea,
1451    totalModules,
1452    areaRelation,
1453    installedModules,
1454    moduleData,
1455    generatordata,
1456    pacProfile,
1457    pacProfile,
1458    generatordata,
1459    radiationData,
1460    monthlyEnergykWh,
1461    anualEnergykWh,
1462    monthlyYf,
1463    anualYf,
1464    inverterPower,
1465    installedPower,
1466    totalArea,
1467    totalModules,
1468    areaRelation,
1469    installedModules,
1470    moduleData,
1471    generatordata,
1472    pacProfile,
1473    pacProfile,
1474    generatordata,
1475    radiationData,
1476    monthlyEnergykWh,
1477    anualEnergykWh,
1478    monthlyYf,
1479    anualYf,
1480    inverterPower,
1481    installedPower,
1482    totalArea,
1483    totalModules,
1484    areaRelation,
1485    installedModules,
1486    moduleData,
1487    generatordata,
1488    pacProfile,
1489    pacProfile,
1490    generatordata,
1491    radiationData,
1492    monthlyEnergykWh,
1493    anualEnergykWh,
1494    monthlyYf,
1495    anualYf,
1496    inverterPower,
1497    installedPower,
1498    totalArea,
1499    totalModules,
1500    areaRelation,
1501    installedModules,
1502    moduleData,
1503    generatordata,
1504    pacProfile,
1505    pacProfile,
1506    generatordata,
1507    radiationData,
1508    monthlyEnergykWh,
1509    anualEnergykWh,
1510    monthlyYf,
1511    anualYf,
1512    inverterPower,
1513    installedPower,
1514    totalArea,
1515    totalModules,
1516    areaRelation,
1517    installedModules,
1518    moduleData,
1519    generatordata,
1520    pacProfile,
1521    pacProfile,
1522    generatordata,
1523    radiationData,
1524    monthlyEnergykWh,
1525    anualEnergykWh,
1526    monthlyYf,
1527    anualYf,
1528    inverterPower,
1529    installedPower,
1530    totalArea,
1531    totalModules,
1532    areaRelation,
1533    installedModules,
1534    moduleData,
1535    generatordata,
1536    pacProfile,
1537    pacProfile,
1538    generatordata,
1539    radiationData,
1540    monthlyEnergykWh,
1541    anualEnergykWh,
1542    monthlyYf,
1543    anualYf,
1544    inverterPower,
1545    installedPower,
1546    totalArea,
1547    totalModules,
1548    areaRelation,
1549    installedModules,
1550    moduleData,
1551    generatordata,
1552    pacProfile,
1553    pacProfile,
1554    generatordata,
1555    radiationData,
1556    monthlyEnergykWh,
1557    anualEnergykWh,
1558    monthlyYf,
1559    anualYf,
1560    inverterPower,
1561    installedPower,
1562    totalArea,
1563    totalModules,
1564    areaRelation,
1565    installedModules,
1566    moduleData,
1567    generatordata,
1568    pacProfile,
1569    pacProfile,
1570    generatordata,
1571    radiationData,
1572    monthlyEnergykWh,
1573    anualEnergykWh,
1574    monthlyYf,
1575    anualYf,
1576    inverterPower,
1577    installedPower,
1578    totalArea,
1579    totalModules,
1580    areaRelation,
1581    installedModules,
1582    moduleData,
1583    generatordata,
1584    pacProfile,
1585    pacProfile,
1586    generatordata,
1587    radiationData,
1588    monthlyEnergykWh,
1589    anualEnergykWh,
1590    monthlyYf,
1591    anualYf,
1592    inverterPower,
1593    installedPower,
1594    totalArea,
1595    totalModules,
1596    areaRelation,
1597    installedModules,
1598    moduleData,
1599    generatordata,
1600    pacProfile,
1601    pacProfile,
1602    generatordata,
1603    radiationData,
1604    monthlyEnergykWh,
1605    anualEnergykWh,
1606    monthlyYf,
1607    anualYf,
1608    inverterPower,
1609    installedPower,
1610    totalArea,
1611    totalModules,
1612    areaRelation,
1613    installedModules,
1614    moduleData,
1615    generatordata,
1616    pacProfile,
1617    pacProfile,
1618    generatordata,
1619    radiationData,
1620    monthlyEnergykWh,
1621    anualEnergykWh,
1622    monthlyYf,
1623    anualYf,
1624    inverterPower,
1625    installedPower,
1626    totalArea,
1627    totalModules,
1628    areaRelation,
1629    installedModules,
1630    moduleData,
1631    generatordata,
1632    pacProfile,
1633    pacProfile,
1634    generatordata,
1635    radiationData,
1636    monthlyEnergykWh,
1637    anualEnergykWh,
1638    monthlyYf,
1639    anualYf,
1640    inverterPower,
1641    installedPower,
1642    totalArea,
1643    totalModules,
1644    areaRelation,
1645    installedModules,
1646    moduleData,
1647    generatordata,
1648    pacProfile,
1649    pacProfile,
1650    generatordata,
1651    radiationData,
1652    monthlyEnergykWh,
1653    anualEnergykWh,
1654    monthlyYf,
1655    anualYf,
1656    inverterPower,
1657    installedPower,
1658    totalArea,
1659    totalModules,
1660    areaRelation,
1661    installedModules,
1662    moduleData,
1663    generatordata,
1664    pacProfile,
1665    pacProfile,
1666    generatordata,
1667    radiationData,
1668    monthlyEnergykWh,
1669    anualEnergykWh,
1670    monthlyYf,
1671    anualYf,
1672    inverterPower,
1673    installedPower,
1674    totalArea,
1675    totalModules,
1676    areaRelation,
1677    installedModules,
1678    moduleData,
1679    generatordata,
1680    pacProfile,
1681    pacProfile,
1682    generatordata,
1683    radiationData,
1684    monthlyEnergykWh,
1685    anualEnergykWh,
1686    monthlyYf,
1687    anualYf,
1688    inverterPower,
1689    installedPower,
1690    totalArea,
1691    totalModules,
1692    areaRelation,
1693    installedModules,
1694    moduleData,
1695    generatordata,
1696    pacProfile,
1697    pacProfile,
1698    generatordata,
1699    radiationData,
1700    monthlyEnergykWh,
1701    anualEnergykWh,
1702    monthlyYf,
1703    anualYf,
1704    inverterPower,
1705    installedPower,
1706    totalArea,
1707    totalModules,
1708    areaRelation,
1709    installedModules,
1710    moduleData,
1711    generatordata,
1712    pacProfile,
1713    pacProfile,
1714    generatordata,
1715    radiationData,
1716    monthlyEnergykWh,
1717    anualEnergykWh,
1718    monthlyYf,
1719    anualYf,
1720    inverterPower,
1721    installedPower,
1722    totalArea,
1723    totalModules,
1724    areaRelation,
1725    installedModules,
1726    moduleData,
1727    generatordata,
1728    pacProfile,
1729    pacProfile,
1730    generatordata,
1731    radiationData,
1732    monthlyEnergykWh,
1733    anualEnergykWh,
1734    monthlyYf,
1735    anualYf,
1736    inverterPower,
1737    installedPower,
1738    totalArea,
1739    totalModules,
1740    areaRelation,
1741    installedModules,
1742    moduleData,
1743    generatordata,

```

```

403     calculationsData: {
404         installedPower,
405         installedModules,
406         inverterPower,
407         AnualEnergykWh,
408         MonthlyEnergykWh,
409         monthlyYf,
410         AnualYf,
411         PacProfile,
412         areaRelation
413     },
414     radiationData
415 };
416 localStorage.setItem('results', JSON.stringify(results));
417 }
418

```

Extracto de código A.8: public/scripts/DataAquisition.js

### A.2.2. GenerateLinks.js

```

1 const minLat = 36.1;
2 const maxLat = 43.63;
3 const minLon = -9.15;
4 const maxLon = 3.06;
5 const latDiff = 0.25;
6 const lonDiff = 0.25
7 const links = [];
8 let data = [];
9 for (let i = minLat; i <= maxLat; i += latDiff) {
10     for (let j = minLon; j <= maxLon; j += lonDiff) {
11         const lon = j.toFixed(2);
12         const lat = i.toFixed(2);
13         links.push(
14             `http://www.adrase.com/adrasemaps/php/monthly_popup.php?lat=${lat}&
15             lon=${lon}&var_tipe=0`

```

```
16    );
17    data.push({
18      lat,
19      lon
20    });
21  }
22}
23
24const textPromises = links.map(async (link) => {
25  const res = await fetch(link, {
26    method: 'GET',
27    mode: 'no-cors'
28  });
29  const text = await res.text();
30  return text;
31});
32Promise.all(textPromises).then(function(values) {
33  let texts = [];
34
35  values.forEach(function(e, i) {
36    if (e.includes('NO HAY DATOS PARA EL PUNTO SELECCIONADO')) {
37      texts.push(null);
38    } else if (e.includes('Valor medio')) {
39      const regex = /[+-]?\d+(\.\d+)?/g;
40      const str = e.split('Valor medio')[1].split('tr')[0];
41      const floats = str.match(regex).map(function(v) {
42        return parseFloat(v);
43      });
44      texts.push(floats);
45    }
46  });
47
48  for (let i = 0; i < texts.length; i++) {
49    data[i].midValues = texts[i];
50  }
51  const filteredData = data.filter((dataElement) => {
```

```

52     return dataElement.midValues != null;
53   );
54   stringData = JSON.stringify(filteredData);
55   console.log(stringData);
56 });
57

```

Extracto de código A.9: public/scripts/generateLinks.js

### A.2.3. index.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <title>Solar calculator</title>
6   <link rel="stylesheet" href="./styles/style.css">
7   <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/bulma/0.6.2/
8     css/bulma.css">
9   <link rel="stylesheet" href="https://use.fontawesome.com/releases
10    /v5.7.2/css/all.css">
11   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
12 </head>
13
14 <body>
15
16   <nav class="navbar is-warning" role="navigation" aria-label="main-navigation">
17     <div class="navbar-brand">
18       <a href="index.html" class="navbar-item title is-5">
19         <span class="icon is-large"><i class="fas
20           fa-lg fa-solar-panel"></i></span> Solarcalc</a
21       >
22     </div>
23     <div class="navbar-end">
24       <div class="navbar-item">
25         <p class="control">

```

```
26         <a href="#" target="_blank" class="has-text-dark title is-5">
27             <span>About</span>
28         </a>
29     </p>
30 </div>
31 <div class="navbar-item">
32     <p class="control">
33         <a href="https://github.com/IonutMorariu/PV-Calculator"
34             target="_blank" class="button is-dark">
35             <span class="icon">
36                 <i class="fab fa-github-alt"></i>
37             </span>
38             <span>Github</span>
39         </a>
40     </p>
41 </div>
42 </div>
43 </nav>
44
45 <div class="container">
46     <div class="columns is-centered">
47         <div class="column is-half">
48             <h3 class="title is-4 has-text-centered">Datos del emplazamiento</h3>
49             <div class="field">
50                 <label class="label" for="address">Direccion</label>
51                 <div class="control has-icons-left has-icons-right">
52                     <input class="input" type="text" placeholder="Calle"
53                         id="address" required value="">
54                     <span class="icon is-small is-left">
55                         <i class="fas fa-map-marker"></i>
56                     </span>
57                 </div>
58             </div>
59             <div class="field is-horizontal">
60                 <div class="field-body">
61                     <div class="field">
```

```
62          <label class="label" for="city">Municipio</label>
63          <div class="control has-icons-left has-icons-right">
64              <input class="input" type="text"
65                  placeholder="Municipio" id="city" required>
66              <span class="icon is-small is-left">
67                  <i class="fas fa-building"></i>
68              </span>
69          </div>
70      </div>
71      <div class="field">
72          <label class="label" for="postal">Código postal</label>
73          <div class="control has-icons-left has-icons-right">
74              <input class="input" type="text"
75                  placeholder="Código postal" id="postal" required>
76              <span class="icon is-small is-left">
77                  <i class="fas fa-hashtag"></i>
78              </span>
79          </div>
80      </div>
81  </div>
82 </div>
83 <h3 class="title is-4">Datos de la casa</h3>
84 <div class="field is-horizontal">
85     <div class="field-body">
86         <div class="field">
87             <label class="label" for="slope">Inclinación
88                 del tejado en grados </label>
89             <div class="control has-icons-left has-icons-right">
90                 <input class="input" type="text"
91                     placeholder="Inclinación aproximada del tejado"
92                     id="slope" required value="20">
93                 <span class="icon is-small is-left">
94                     <i class="fas fa-home"></i>
95                 </span>
96             </div>
97         </div>
```

```
98         <div class="field">
99             <label class="label" for="area">rea del tejado
100                en metros cuadrados</label>
101
102             <div class="control has-icons-left has-icons-right">
103                 <input class="input" type="text"
104                   placeholder="rea del tejado" id="area"
105                   required value="23">
106
107                 <span class="icon is-small is-left">
108                     <i class="fas fa-calculator"></i>
109                 </span>
110
111             </div>
112
113         </div>
114
115     <div class="field is-horizontal">
116
117         <div class="field-body">
118
119             <div class="field">
120
121                 <label for="" class="label">Orientación</label>
122
123                 <p class="control has-icons-left">
124
125                     <input class="input" type="number"
126                       placeholder="Orientación" id="orientation"
127                       required value="0">
128
129
130                     <!-- <span class="select">
131                         <select id="orientation">
132
133                             <option disabled>Orientación</option>
134
135                             <option value="180">Norte (N)</option>
136
137                             <option value="225">Noreste (NE)</option>
138
139                             <option value="270">Este (E)</option>
140
141                             <option value="315">Sureste (SE)</option>
142
143                             <option selected value="0">Sur (S)</option>
144
145                             <option value="45">Suroeste (SW)</option>
146
147                             <option value="90">Oeste (W)</option>
148
149                             <option value="135">Noroeste (NW)</option>
150
151                         </select>
152
153                     </span> -->
```

```
134             <span class="icon is-small is-left">
135                 <i class="fas fa-compass"></i>
136             </span>
137         </p>
138     </div>
139
140     <div class="field">
141         <label for="" class="label">Nivel de suciedad</label>
142         <p class="control has-icons-left">
143             <span class="select">
144                 <select id="dirt-level">
145                     <option disabled>Suciedad</option>
146                     <option value="CLEAN">Limpio</option>
147                     <option selected value="LOW">
148                         Nivel bajo
149                     </option>
150                     <option value="MID">Nivel medio</option>
151                     <option value="HIGH">Nivel alto</option>
152                 </select>
153             </span>
154             <span class="icon is-small is-left">
155                 <i class="fas fa-broom"></i>
156             </span>
157         </p>
158     </div>
159
160 </div>
161 <!-- <div class="field">
162     <label class="checkbox">
163         <input id="apply-dirt" type="checkbox" checked
164             aria-checked="true">
165             Aplicar suciedad y angulo de incidencia (solo para pruebas)
166     </label>
167 </div> -->
168 <div class="control has-text-centered">
169     <a class="button is-warning is-centered" id="button-get-coords">
```

```
170             <span class="icon">
171                 <i class="fas fa-paper-plane"></i>
172             </span>
173             <span>Enviar</span>
174         </a>
175     </div>
176 </div>
177 <!--
178     <div>
179         <table class="test-data" border="1">
180             <thead>
181                 <tr>
182                     <th>Month</th>
183                     <th>G0d</th>
184                     <th>Bd0</th>
185                     <th>Dd0</th>
186                     <th>Gefd</th>
187                     <th>Defd</th>
188                     <th>Befd</th>
189                     <th>Eac</th>
190                     <th>Edc</th>
191                     <th>Yf</th>
192                 </tr>
193             </thead>
194             <tbody id="radiation-data"></tbody>
195         </table>
196     </div> -->
197 </body>
198 <script src=".//scripts/apiKeys.js"></script>
199 <script src=".//scripts/dataAquisition.js"></script>
200
201
202 </html>
```

## Extracto de código A.10: public/index.html

#### A.2.4. Results.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <title>Resultados</title>
5          <link rel="stylesheet" href=".//styles/style.css" />
6          <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/
7              bulma/0.6.2/css/bulma.css" />
8          <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.2/css/all.css" />
9          <script src="https://cdn.jsdelivr.net/npm/chart.js@2.8.0/dist/Chart.min.js"></script>
10         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
11     </head>
12
13 <body>
14     <nav class="navbar is-warning" role="navigation" aria-label="main-navigation">
15         <div class="navbar-brand">
16             <a href="index.html" class="navbar-item title is-5">
17                 <span class="icon is-large"><i class="fas fa-lg
18                     fa-solar-panel"></i></span> Solarcalc</a>
19             >
20         </div>
21         <div class="navbar-end">
22             <div class="navbar-item"></div>
23             <div class="navbar-item">
24                 <p class="control">
25                     <a href="https://github.com/IonutMorariu/PV-Calculator"
26                         target="_blank" class="button is-dark">
27                         <span class="icon">
28                             <i class="fab fa-github-alt"></i>
29                         </span>
30                         <span>Github</span>
31                     </a>
32                 </p>
33             </div>
34         </div>
35     </nav>
36     <div class="container">
37         <div class="columns">
38             <div class="column is-3">
39                 <div class="card">
40                     <div class="card-image">
41                         <img alt="Solar panel icon" data-bbox="117 117 188 138" data-label="Image"/>
42                     <div class="mask">
43                         <div class="content">
44                             <strong>Solarcalc</strong>
45                             <div>
46                                 <span>Solar panel calculator</span>
47                                 <span>Calculate the energy output of your solar panels</span>
48                             </div>
49                         </div>
50                     </div>
51                 </div>
52             </div>
53             <div class="column is-3">
54                 <div class="card">
55                     <div class="card-image">
56                         <img alt="Calculator icon" data-bbox="117 218 188 239" data-label="Image"/>
57                     <div class="mask">
58                         <div class="content">
59                             <strong>Solarcalc</strong>
60                             <div>
61                                 <span>Solar panel calculator</span>
62                                 <span>Calculate the energy output of your solar panels</span>
63                             </div>
64                         </div>
65                     </div>
66                 </div>
67             </div>
68             <div class="column is-3">
69                 <div class="card">
70                     <div class="card-image">
71                         <img alt="GitHub icon" data-bbox="117 318 188 339" data-label="Image"/>
72                     <div class="mask">
73                         <div class="content">
74                             <strong>Solarcalc</strong>
75                             <div>
76                                 <span>Solar panel calculator</span>
77                                 <span>Calculate the energy output of your solar panels</span>
78                             </div>
79                         </div>
80                     </div>
81                 </div>
82             </div>
83         </div>
84     </div>
85 
```

```
32          </p>
33      </div>
34  </div>
35 </nav>
36 <div class="section">
37   <div class="columns is-centered">
38     <div class="column is-half">
39       <div class="box">
40         <h1 class="title is-4 has-text-centered">
41           Resultados del clculo</h1>
42         <h2 class="title is-5">Datos del emplazamiento:</h2>
43         <ul class="list">
44           <div class="data-grid">
45             <li><span class="has-text-weight-semibold">
46               Latitud:</span> <span id="latitude"></span></li>
47             <li><span class="has-text-weight-semibold">
48               Longitud:</span> <span id="longitude"></span></li>
49             <li><span class="has-text-weight-semibold">
50               Radiacin global media:</span>
51               <span id="global-radiation"></span></li>
52             <li><span class="has-text-weight-semibold">
53               Superficie:</span> <span id="area"></span></li>
54             <li><span class="has-text-weight-semibold">
55               Inclinacin:</span>
56               <span id="slope"></span></li>
57             <li><span class="has-text-weight-semibold">
58               Orientacin:</span>
59               <span id="orientation"></span></li>
60           </div>
61         </ul>
62         <hr style="margin: 0 0 3rem;" />
63         <h2 class="title is-5">Datos de la instalacion calculada:</h2>
64         <ul class="list">
65           <div class="columns">
66             <div class="column is-one-half">
67               <li><span class="has-text-weight-semibold">
```

```
68     Numero de modulos:</span>
69     <span id="installed-modules"></span></li>
70     <li><span class="has-text-weight-semibold">
71         Potencia minima del inversor:</span>
72         <span id="inverter-power"></span></li>
73     </div>
74     <div class="column is-one-half">
75         <li><span class="has-text-weight-semibold">
76             Potencia instalada:</span>
77             <span id="installed-power"></span></li>
78         <li><span class="has-text-weight-semibold">
79             Energia anual producida:</span>
80             <span id="anual-energy"></span></li>
81         </div>
82     </div>
83 </ul>
84 <hr style="margin: 0 0 1rem;" />
85
86 <p class="content is-small">
87     Los calculos han sido realizados en
88     base al módulo de potencia nominal
89     320W del fabricante JINKO SOLAR, modelo JKM320PP
90     <a target="_blank" href="https://autosolar.es/pdf
91     /Ficha-Tecnica-Jinko-Solar-305-320W.pdf">ENLACE</a>
92     <br />
93     <!-- Esta aplicación es el
94     resultado del Trabajo de Fin de Grado de Ionut Morariu.
95     Puede leer todo el proceso del cálculo que se ha
96     seguido para conseguir los resultados en este documento.
97     <a target="_blank" href="https://github.com/
98     IonutMorariu/pv-calculator">ENLACE</a> -->
99     </p>
100    </div>
101    <h2 class="title is-5 has-text-centered">
102        Gráficos adicionales:</h2>
103
```

```

104         <canvas id="chart-1" width="400" height="200"></canvas>
105         <canvas id="chart-2" width="400" height="200"></canvas>
106         <canvas id="chart-3" width="400" height="200"></canvas>
107     </div>
108 </div>
109 </div>
110 </body>
111 <script src="bling.js"></script>
112 <script src="renderResults.js"></script>
113 </html>
114

```

Extracto de código A.11: public/results.html

### A.2.5. renderResults.js

```

1 const results = JSON.parse(localStorage['results'])
2 console.log(results)
3 //Placement data
4 $('#area').innerHTML = results.placementData.area + ' m2</sup>'
5 $('#latitude').textContent = results.placementData.location.lat.toFixed(2)
6 $('#longitude').textContent = results.placementData.location.long.toFixed(2)
7 $('#orientation').textContent = results.placementData.orientation + ' '
8 $('#slope').textContent = results.placementData.slope + ' '
9 $('#global-radiation').innerHTML =
10 (
11     results.radiationData.meanValues.reduce((prev, next) => {
12         return (prev += next.meanGR)
13     }, 0) / 12
14 ).toFixed(2) + ' kWh/m2</sup>'
15
16 //Calculation data
17
18 $('#installed-modules').textContent = results.calculationsData.installedModules
19 $('#installed-power').textContent = results.calculationsData.installedPower + ' W'
20 $('#inverter-power').textContent = results.calculationsData.inverterPower + ' W'

```

```
21 $( '#anual-energy' ).textContent =
22   ((results.calculationsData.AnualEnergykWh / 1000) *
23     results.calculationsData.areaRelation).toFixed(2) + ' MWh'
24
25 const ctx1 = document.getElementById('chart-1').getContext('2d')
26 const ctx2 = document.getElementById('chart-2').getContext('2d')
27 const ctx3 = document.getElementById('chart-3').getContext('2d')
28
29 const myChart1 = new Chart(ctx1, {
30   type: 'line',
31   data: {
32     labels: ['Enero', 'Febrero', 'Marzo', 'Abril',
33       'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre',
34       'Octubre', 'Noviembre', 'Diciembre'],
35     datasets: [
36       {
37         label: 'Energia mensual en kWh',
38         data: results.calculationsData.MonthlyEnergykWh.map(val =>
39           (val * results.calculationsData.areaRelation).toFixed(2)),
40         backgroundColor: ['#ff76751f'],
41         borderColor: ['#ff7675'],
42         lineTension: 0,
43       },
44     ],
45   },
46   options: {
47     responsive: true,
48     scales: {
49       yAxes: [
50         {
51           ticks: {
52             beginAtZero: true,
53           },
54         },
55       ],
56     },
57   },
58 }
```

```
57     },
58 })
59 const myChart2 = new Chart(ctx2, {
60   type: 'line',
61   data: {
62     labels: ['Enero', 'Febrero', 'Marzo',
63       'Abril', 'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre',
64       'Octubre', 'Noviembre', 'Diciembre'],
65     datasets: [
66       {
67         label: 'Productividad diaria media',
68         data: results.calculationsData.monthlyYf.map(val =>
69           (val / 30).toFixed(2)),
70         backgroundColor: ['#0984e31f'],
71         borderColor: ['#0984e3'],
72         lineTension: 0,
73       },
74     ],
75   },
76   options: {
77     responsive: true,
78     scales: {
79       yAxes: [
80         {
81           ticks: {
82             beginAtZero: true,
83           },
84         },
85       ],
86     },
87   },
88 })
89 console.log(results.calculationsData.PacProfile.map(profile => Math.max(...profile)))
90 const myChart3 = new Chart(ctx3, {
91   type: 'line',
92   data: {
```

```
93     labels: ['Enero', 'Febrero', 'Marzo', 'Abril',
94     'Mayo', 'Junio', 'Julio', 'Agosto', 'Septiembre', 'Octubre',
95     'Noviembre', 'Diciembre'],
96     datasets: [
97     {
98         label: 'Potencia Mxima diaria en W',
99         data: results.calculationsData.PacProfile.map(profile =>
100             (Math.max(...profile) * results.calculationsData.areaRelation)
101             .toFixed(2)),
102         backgroundColor: ['#00b8941f'],
103         borderColor: ['#00b894'],
104         lineTension: 0,
105     },
106     ],
107 },
108 options: {
109     responsive: true,
110     scales: {
111         yAxes: [
112             {
113                 ticks: {
114                     beginAtZero: true,
115                 },
116             },
117         ],
118     },
119 },
120 })
```

Extracto de código A.12: public/renderResults.js

# Bibliografía

- [1] Unión Española Fotovoltaica. 2019. Informe anual 2019: El sector fotovoltaico impulsor de la transición energética. [https://unef.es/wp-content/uploads/dlm\\_uploads/2019/09/memoria\\_unef\\_2019-web.pdf](https://unef.es/wp-content/uploads/dlm_uploads/2019/09/memoria_unef_2019-web.pdf).
- [2] PVsyst. PVsyst SA. <https://www.pvsyst.com/download-pvsyst/>.
- [3] CalculationSolar. 2014. <http://www.calculationsolar.com/>.
- [4] SISIFO.Instituto de Energía Solar. <https://www.sisifo.info/es/default>.
- [5] PVGIS. European Commission, Joint Research Centre. <https://re.jrc.ec.europa.eu/pvg-tools/es/tools.html>.
- [6] System Advisor Model. NREL, US Department of Energy <https://sam.nrel.gov/>.
- [7] Oscar Perpiñán (2012). solaR: Solar Radiation and Photovoltaic Systems with R, Journal of Statistical Software, 50(9), 1-32. <http://www.jstatsoft.org/v50/i09/>.
- [8] Perpiñán, O. 2018. Energía Solar Fotovoltaica. <http://oscarperpinan.github.io/esf/>.
- [9] B. Y. H. Liu y R. C. Jordan. "The interrelationship and characteristic distribution of direct,diffuse, and total solar radiation". *Solar Energy* 4 (1960), págs. 1-19
- [10] Estimating average daytime and daily temperature profiles within Europe. Thomas A.Huldm Marcel Šúri, Ewan D.Dunlop, Fabio Micale 2006 <https://www.sciencedirect.com/science/article/abs/pii/S1364815205001593>
- [11] NodeJS ORG. <https://nodejs.org/es/about/>.