

Desarrollo de una aplicación web para la configuración de instalaciones de energía solar fotovoltaica.

Autor: [Aurélien Camilleri](#)

Tutores: [Ángel Marqués Mateu](#), ETSIGT, Universitat Politècnica de València

[José Ángel Garrido Saraso](#), Departamento técnico I+D+i, Techno Sun

Curso: 2020/2021

Introducción

La idea general de este proyecto es crear una plataforma web que ayude a configurar y diseñar instalaciones solares fotovoltaicas. Esta plataforma será utilizada, en un principio, por los técnicos e ingenieros de TECHNOSUN, y luego, si todo funciona perfectamente, la plataforma podrá ser abierta a todo el mundo.

Objetivos

Los objetivos del proyecto son los siguientes:

- Crear un visor cartográfico con ortofotos en el que pueda hacer clic para seleccionar la longitud y la latitud de los equipos a instalar,
- Implementar los cálculos electro-técnicos para los equipos fotovoltaicos en Python,
- Utilizar datos astronómicos (orientación del sol, ...) y meteorológicos (radiación solar, ...) precisos,
- Crear una base de datos con el máximo número de paneles solares e inversores posibles (no necesariamente sólo los propuestos por TECHNOSUN),
- Que el proyecto sea exclusivamente gratis.

Metodología

Para iniciar el proyecto, primero se codificaron los cálculos electrónicos a partir de la documentación que aportó la empresa. Con estos PDFs, se creó una librería de funciones de cálculos electrónicos fotovoltaicos, en Python orientado objetos. Se ha elegido Python orientado objetos porque es más fácil llamar desde otros códigos y otros lenguajes de programación. Las variables y parámetros de entrada son: Longitud, Latitud, Características del panel solar y de los equipos fotovoltaicos (inversores, reguladores, variadores de frecuencia). Con consultas, se calculará la temperatura. La lista de materiales se podrá ampliar durante el desarrollo del software que continuará después de este TFM, en colaboración con la empresa.

Para trabajar con datos precisos, para la temperatura se utilizan las APIs de PVGIS. En el caso de la altitud, se utiliza la API para hacer la consulta a OpenTopoData. En ambos casos, resultan datos muy precisos y la respuesta es muy rápida. La fiabilidad de los datos consultados es buena y permite, a posteriori, cálculos muy precisos.

Inicialmente, no se disponía de coordenadas reales para probar las funciones y lo primero que se hizo fue desarrollar el visor.

La implementación del visor cartográfico se ha realizado utilizando los lenguajes de programación HTML y JavaScript. Posteriormente, se han utilizado los mapas Leaflet. Son mapas en acceso y código libres. Al mismo tiempo, se van agregando y creando mecanismos para poder obtener las coordenadas del punto que se ha clicado.

Una vez obtenidas las coordenadas del punto, el siguiente paso, ha sido enlazar con la librería Python que contienen los métodos que procesan los cálculos para dichas coordenadas. Es el lenguaje de programación JavaScript, el que envía los datos de las coordenadas tipo de modulo fotovoltaico, tipo de inversor (etc...) a una capa intermedia de software llamada Django.

Django es un framework de lenguaje Python. Se ha elegido porque ya se tenía experiencia y porque ha demostrado ser eficiente, además que continua con la filosofía de software gratis. Es Django, el que lleva el peso de poder actuar directamente con los métodos implementados en Python.

A partir de los parámetros pasados por Django, los métodos de Python calculan y elaboran las respuestas.

Una vez que está bien implementado, se realizó la siguiente prueba:

- Recoger coordenadas desde el mapa,
- Pasarlas a las librerías Python mediante la intervención de Django,
- Inicialmente, se ha utilizado la función para determinar la altitud a partir de las coordenadas,
- Devolver la altitud al código JavaScript pasando por la capa Django,
- Con el fin de visualizar los datos en la pantalla.

Este paso ha sido la clave para el posterior desarrollo de todas las consultas que se realizan entre las tres piezas software (JavaScript, Django y Python).

Ahora se tiene que crear la base de datos para permitir la elección del material fotovoltaico (módulos e inversores). Se ha decidido crear dos códigos Python, uno para objetos tipo modulo fotovoltaico y otro para los objetos tipo inversores. Al ejecutar, estos códigos rellenan la base de datos de la empresa, con los objetos equipos y módulos, y sus respectivos atributos técnicos. El formato utilizado es JSON, porque se ha convertido en el formato base de intercambio de datos tanto entre máquinas como entre máquinas y usuarios.

Siguiendo con el desarrollo, se pasa al diseño de la interfaz gráfica con más profundidad para mejorar la vista de la página y sus funcionalidades. Se añade más campos y controles para elegir el módulo y el inversor. Luego, se repite el proceso para mostrar en la página web, todos los campos de interés para los usuarios de la empresa.

Un toque final para embellecer la página con los colores del logo de TECHNOSUN y la paleta de colores que lo acompañan.

Resultados

El resultado final es una página complementada funcional que contiene todos los datos que define los componentes del sistema fotovoltaico. Y lo más importante del software es como relaciona las coordenadas cartográficas, determina las temperaturas extremas y medias para dichas coordenadas y finalmente determina el comportamiento exacto del material fotovoltaico para garantizar la fiabilidad y durabilidad del sistema o de la instalación.

Los campos los más interesantes para los usuarios de TECHNOSUN son los campos de la línea N° Module Serie porque dan la estimación final. Pasar por la página web que se ha hecho hace ganar tiempo: no hay que utilizar la calculadora, ni un tablero Excel complejo. Solo enseñar la posición, las marcas de los equipos y el resultado ya está aquí abajo. Además de todo, esta página permite evitar los errores de cálculo y descuidados.

Para que la página sea interesante y fácil de usar, he creado diferentes gadgets como:

- La barra de búsqueda con un geocoding para permitir al usuario de buscar un lugar en particular,
- El botón de localización que permite seleccionar automáticamente tu posición misma (también, te indica la precisión),
- El logo TECHNOSUN contiene un enlace que te dirige directamente en la página oficial de TECHNOSUN,
- Los botones + y -, arriba a la izquierda del mapa, funcionan para ampliarlo y reducirlo,
- Si se reduce el tamaño de la página, el mapa cambia de sitio.

Conclusiones

El proyecto era complejo e interesante. Gracias a las habilidades desarrolladas durante el curso y a la ayuda de los tutores, el proyecto fue un éxito. Aunque hemos encontrado problemas, como la API de PVGIS que no se puede llamar desde JavaScript sino desde Python, ahora resulta de gran ayuda para los técnicos de TECHNOSUN. Todavía se tiene que verificar si cumplimos todos los objetivos:

- Crear un visor cartográfico con ortofotos en el que pueda hacer clic para seleccionar la longitud y la latitud de los equipos a instalar,
- Hecho.** Tenemos un visor operacional que permite hacer clic para recoger las coordenadas.

- Crear una base de datos con el máximo número de paneles solares e inversores posibles (no necesariamente sólo los propuestos por TECHNOSUN),
- No totalmente hecho.** Tenemos una base de datos completas y operacional. Sin embargo, solo contiene los datos de los equipos usados por la empresa. Se tiene, en el futuro, ampliar la base de datos.

- Implementar los cálculos electro-técnicos para los equipos fotovoltaicos en Python,
- Hecho.** Tenemos una gran librería de funciones usando los diferentes cálculos electro-técnicos. Puede ser necesario añadir otras funciones más adelante para realizar otros cálculos.

- Utilizar datos astronómicos (orientación del sol, ...) y meteorológicos (radiación solar, ...) precisos,
- Hecho.** Los datos que no eran seguros fueron sustituidos por datos de las APIs.

- Que sea exclusivamente gratis (al menos en primer lugar).

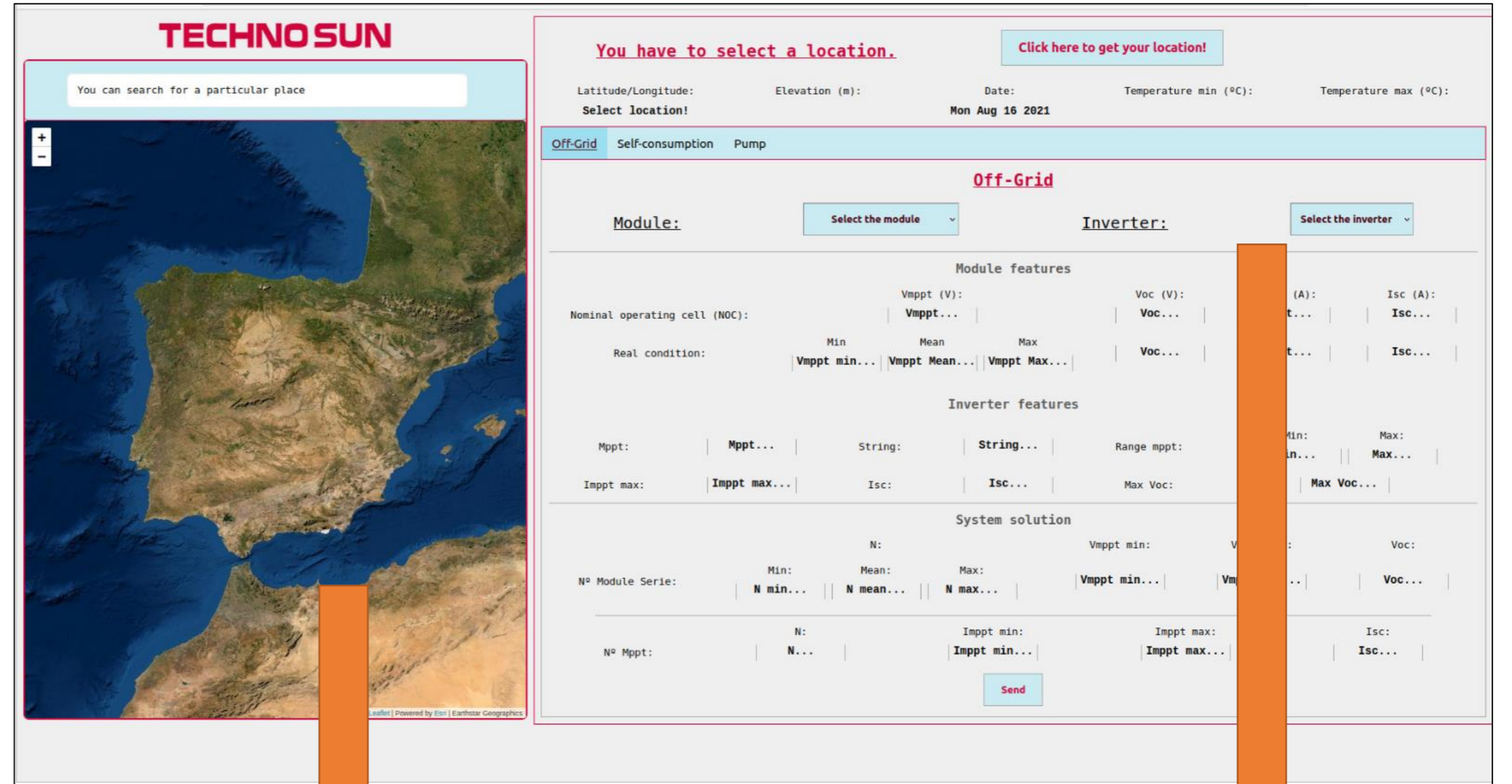
- Hecho.** Tenemos un resultado bien potente que puede competir con aplicaciones con costosas económicamente. Los datos que no eran seguros fueron sustituidos por datos de las APIs.

Aplicaciones

Esta página web ya sirve a los técnicos de la empresa TECHNOSUN. Por ejemplo, en el servicio al cliente, el técnico, mientras habla por teléfono, puede introducir los datos del cliente en la página para tener un resultado inmediato sobre las capacidades del dispositivo del cliente.

Esta página permite salvar mucho tiempo de cálculo y evitar los errores de cálculo y descuidados. Entonces permite ganar una gran seguridad y ganar en eficiencia en general. La página debería permitir ser más competitivo y potente que las empresas que no tienen este tipo de herramienta.

Desarrollo



`getAltitude(technosun)` `readTemperature(technosun)`

Latitude/Longitude:	Elevation (m):	Date:	Temperature min (°C):	Temperature max (°C):
39.481, -0.337	8.247	Mon Aug 16 2021	5.15	37.89

`readModuleData(technosun)`

`readInverterData(technosun)`

Module Features				
Nominal operating cell (NOC):	Vmppt (V):	Voc (V):	Imppt (A):	Isc (A):
48.5	49.2	49.2	10	10.52
Real condition:				
Min:	Mean:	Max:	Min:	Max:
29.54	33.48	37.43	46.13	10.85

Inverter features				
Mppt:	2	String:	1	Range mppt:
Imppt max:	10	Zak:	14	Max Voc:
				999

`readRealModuleData(technosun)`

Module Features				
Nominal operating cell (NOC):	Vmppt (V):	Voc (V):	Imppt (A):	Isc (A):
48.5	49.2	49.2	10	10.52
Real condition:				
Min:	Mean:	Max:	Min:	Max:
29.54	33.48	37.43	46.13	10.85

`system_solution()`

System solution				
N° Module Serie:	Min:	Mean:	Max:	Vmppt min:
9	11	13	324.9	411.08
				507.38
N° Mppt:	N:	Imppt min:	Imppt max:	Isc:
1	1	10	10	14