
This is the **published version** of the bachelor thesis:

Martinez Abril, Javier; Martí Escalé, Ramon, dir. Meteo : Módulo de información meteorológica para la plataforma Wattwin. 2021. (958 Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/248538>

under the terms of the  license

Meteo: Módulo de información meteorológica para la plataforma Wattwin

Javier Martínez Abril

Resumen– La empresa SIIE es la propietaria de la plataforma Wattwin, una solución digital para diseñar, vender y gestionar instalaciones fotovoltaicas en la que se utilizaba un sistema poco fiable para el cálculo de la radiación y producción solar en una ubicación, las tablas CTE. Para solucionar ese problema inició el desarrollo de este proyecto, un módulo que permita obtener y presentar datos meteorológicos de una ubicación extrayendo los datos de una base de datos fiable, PVGIS.

El proyecto se inició adaptando el nuevo mecanismo a los otros módulos y desarrollando la vista de los usuarios estándar permitiéndoles seleccionar una ubicación y obtener información meteorológica. Seguidamente se desarrolló la vista de los usuarios administradores permitiéndoles gestionar los datos almacenados en caché. El módulo se encuentra en la versión de desarrolladores de la plataforma y será lanzado en producción en la siguiente actualización.

Palabras clave– SIIE, Wattwin, PVGIS, base de datos, meteorológica, desarrollo

Abstract– The company SIIE is the owner of the Wattwin platform, a digital solution for designing, selling and managing photovoltaic installations which previously had an unreliable system to calculate the radiation and solar production at a given location, known as CTE tables. To solve that issue, the company initiated the development of this tool, a module that allows users to obtain and register meteorological data of a certain location, extracting the data from a reliable database, PVGIS.

The project started by adapting the new mechanism to the existing modules, as well as developing a new user interface to simplify the access to requested meteorological data in a given location. Subsequently, the administrator users' view was developed, allowing them to manage cached data. The module is in developing phase on the platform and will be released in production in future updates.

Keywords– SIIE, Wattwin, PVGIS, database, meteorological, develop

1 INTRODUCCIÓN - CONTEXTO DEL TRABAJO

Las energías renovables son un tema puntero hoy en día, ya que existe la necesidad de utilizar este tipo de energías debido a que son mucho menos contaminantes y además su existencia no es limitada. Actualmente, se dispone de diferentes tipos de energías renovables como energía hidráulica, energía eólica o la que se comentará a lo largo de este documento: energía solar fotovoltaica.

La empresa SIIE o Servicios Integrales a Infraestructuras Energéticas [1], una consultoría con sede en Barcelona que

- E-mail de contacto: javier.martinezab@e-campus.uab.cat
- Mención realizada: Tecnologías de la Información
- Trabajo tutorizado por: Ramon Martí (departament)
- Curso 2020/21

brinda soluciones personalizadas a sus clientes y socios tanto en el uso de tecnologías de almacenamiento y de nueva generación, como en la subcontratación de servicios operativos y de gestión, así como de servicios de diseño, permisos y operación para sistemas fotovoltaicos solares.

SIIE detectó que no existía una aplicación o plataforma que centralice todos los complejos procesos que se deben realizar para la instalación y explotación de los recursos solares, tanto para particulares como para empresas. Es por ello que decidieron crear una plataforma donde centralizar el diseño, la planificación, la facturación, los trámites legales, etc. con la finalidad de facilitar todos esos procesos.

De esa idea nació Wattwin [2], una solución digital para el diseño, venta y gestión de instalaciones fotovoltaicas, en la que se ofrecen diferentes funciones como dibujar, calcular, cotizar y gestionar los proyectos relacionados con el autoconsumo solar.

La plataforma no contaba con la funcionalidad de ofrecer

información fiable sobre datos meteorológicos de un punto en concreto, sino que utilizaban unas tablas CTE (Código Técnico de Edificación) con las que calcular la producción solar y la radiación en un punto. Por este motivo se agregó un módulo que permitiese obtener y presentar datos meteorológicos de una ubicación a partir de un clic en el mapa, añadiendo la dirección de un lugar concreto o añadiendo la latitud y longitud de un punto y extraer esos datos de una base de datos fiable como PVGIS (“Photovoltaic Geographical Information System”) [3].

Por otra parte, esos datos iban a ser almacenados en la base de datos con la intención de reducir las llamadas a esa base de datos externa, por ese motivo se decidió crear un portal de administración donde los usuarios administradores de Wattwin pudiesen purgar la base de datos en el intervalo de fechas que ellos decidiesen.

A lo largo de este documento se explicarán los objetivos y requisitos de este módulo, se comentará el estado del arte, se explicará la metodología que se siguió para su correcto desarrollo y finalmente, se mostrará la planificación que se ha seguido a lo largo de esos meses.

2 OBJETIVOS

El objetivo de este proyecto es agregar a la plataforma Wattwin un módulo donde poder obtener y presentar información meteorológica de un punto. Para poder indicar ese punto se desea hacer de tres posibles maneras: haciendo clic en el mapa y que automáticamente detecte las coordenadas de ese punto, indicar una dirección y que automáticamente busque las coordenadas de esa dirección, o indicar una latitud y una longitud. Para la visualización de estos datos, el usuario debe poder consultar la radiación, la temperatura, la humedad, la velocidad del viento y la producción en diferentes periodos y, además, poder editar los datos para obtener la producción con una disposición personalizada. Por otra parte, el usuario estándar también debe ser capaz de asignar una ubicación a una planta fotovoltaica la cual podrá ajustar una radiación solar total.

El usuario administrador debe realizar otras funciones más enfocadas a la administración como consultar el número de llamadas que se han realizado a PVGIS, comprobar el número de direcciones que hay almacenadas en ese momento en memoria caché, borrar/vaciar esa memoria y finalmente comprobar la relación de aciertos y fallos del cacheado.

A continuación, se mostrará un listado, a grandes rasgos, de los objetivos de este proyecto.

- **Integrar Meteo con otros módulos ya existentes.** Con la creación del módulo Meteo se debe adaptar el comportamiento de los otros módulos que funcionan realizando sus cálculos a través de las tablas CTE para que funcionen con el nuevo módulo. Por otra parte se debe permitir a los usuarios ajustar la radiación solar total a una planta FV extrayendo también los datos de este nuevo módulo.
 - **Ajustar radiación solar total a planta FV.** Entre las diferentes plantas fotovoltaicas ya existentes en la plataforma se debe permitir la posibilidad de ajustar una radiación solar total a una planta en función de sus coordenadas.

- **Adaptar nuevo formato de radiación.** Al realizar dichos cambios también se debe ajustar el funcionamiento de los diferentes módulos para que a partir de ese momento usen el nuevo formato ya que la lógica actual, las tablas CTE, deben desaparecer.
- **Crear y Configurar mapa de Meteo.** Lo primero que deben ver los usuarios al entrar al módulo es un mapa de España al que poder definir una ubicación de diferentes maneras y esa ubicación debe quedar claramente reflejada en el mapa.
 - **Consultar mapa.** Al entrar al módulo debe aparecer un mapa de la API de Google Maps. Previamente a que el usuario indique un punto en el mapa, se mostrará el mapa completo de España.
 - **Definir y consultar ubicación.** Definir una ubicación en el mapa ya sea haciendo clic, introduciendo una dirección o coordenadas.
- **Representar y Editar los datos.** Para que los usuarios entiendan fácilmente los datos, estos se deben representar de manera que sea cómoda para el usuario. Por otra parte deben tener la opción de editar la disposición óptima y consultar datos meteorológicos con la nueva disposición.
 - **Representar datos.** Los diferentes datos meteorológicos obtenidos de la API PVGIS se deben poder visualizar como un resumen o gráficos.
 - **Editar los datos representados.** Se debe ofrecer la posibilidad de personalizar la inclinación y orientación de las placas fotovoltaicas y obtener datos meteorológicos con esa disposición.
- **Crear y configurar portal de administración de Meteo.** El módulo genera grandes cantidades de datos con lo que se debe ofrecer la posibilidad de purgar esos datos en el intervalo que los administradores de Wattwin deseen. Por otra parte se debe ofrecer la posibilidad de comprobar si el cacheado de los datos esta funcionando correctamente.
 - **Comprobar volumen caché.** Los administradores de Wattwin deben tener la opción de comprobar el número de direcciones se encuentran almacenadas en la memoria caché.
 - **Borrar caché.** Se debe ofrecer a los administradores de Wattwin la posibilidad de vaciar la caché.
 - **Consultar llamadas PVGIS.** Ofrecer también la posibilidad de consultar el número de llamadas que se han realizado a la API mensual o diariamente.
 - **Contar aciertos y fallos de cacheado.** Con la finalidad de comprobar si está siendo útil el cacheado de la información contar el número de aciertos y el número de fallos en las peticiones.
 - **Representar relación aciertos/fallos.** Representar los aciertos y fallos en un gráfico para su fácil entendimiento.

- **Definir y Ejecutar pruebas finales y Corregir errores.** Esta tarea se centra en someter las nuevas funcionalidades a diferentes pruebas y casos con el fin de encontrar el mayor número de errores posibles. Solucionar esos errores y volver a someter las funcionalidades a diferentes pruebas.
 - **Definir pruebas.** Definir un conjunto de pruebas en diferentes escenarios.
 - **Ejecutar pruebas.** Someter las nuevas funcionalidades al conjunto de pruebas definido previamente con el fin de encontrar el mayor número de errores posibles.
 - **Corregir errores.** Solucionar todos los errores encontrados.

Por lo que respecta a la prioridad, se han ordenado los objetivos iniciando por el más prioritario, ya que la empresa solicitó que inicialmente se realizara la integración de Meteo con otros módulos ya existentes, para que posteriormente se siguiera con la creación y configuración del mapa de Meteo y seguidamente con la representación y edición de datos. Finalmente se creó y configuró el portal de administración de Meteo. Una vez fueron desarrolladas y finalizadas cada una de las funcionalidades, estas se sometieron a un conjunto de pruebas para comprobar que todo funcionaba como debería.

3 ESTADO DEL ARTE

Wattwin es una plataforma web propiedad de la empresa SIIE que permite diseñar, vender y gestionar instalaciones fotovoltaicas. Actualmente la plataforma se encuentra en funcionamiento y cuenta con diferentes tipos de clientes, pero todavía se encuentra en desarrollo, ya que poco a poco aparecen nuevas funcionalidades que la empresa desea añadir. Por este motivo se deseaba añadir el módulo Meteo, ya que la plataforma no contaba con una sección donde obtener información meteorológica fiable en un punto en concreto. En algunos módulos se hace referencia a una zona climática según el CTE (Código Técnico de Edificación) y a la radiación solar kWh/h de un punto definido, datos que, hoy en día, no son del todo fiables debido a que no se actualizan en tiempo real.

Con la creación de este módulo se agruparon todos esos datos, pero esta vez extrayéndolos de una base de datos fiable (PVGIS) y es posible mostrarlos de diferentes maneras desde históricos anuales hasta detalles horarios, ya sea de forma textual o en forma de gráficos, además de haber sido integrado con otros módulos ya existentes.

Junto a este módulo se ha añadido una funcionalidad para que los administradores de Wattwin puedan gestionar la memoria caché y comprobar el número de llamadas que se ha realizado a la API así como comprobar si el cacheado de los datos esta funcionando de manera satisfactoria. Como se ha comentado, la plataforma está en funcionamiento y ha sido desarrollada con software muy puntero en el mercado, el cual se describe a continuación:

- **Angular [5]:** un framework de desarrollo para JavaScript, creado para facilitar la creación y programación de aplicaciones web.

- **TypeScript [9]:** una adaptación de JavaScript para convertirlo de tipado estático.
- **Pug [7]:** un template engine que mejora la legibilidad del código.
- **Sass [8]:** un lenguaje que permite agregar características no existentes en CSS, como variables, funciones, etc.
- **LoopBack [6]:** un framework de Node.js que permite crear rápidamente API y micro-servicios compuestos desde el back end.
- **MongoDB [4]:** una base de datos NoSQL orientada a documentos.
- **Bitbucket [10]:** un servicio de alojamiento basado en web.

4 METODOLOGÍA

Para el desarrollo de este proyecto se realizaron semanalmente pequeñas reuniones con diferentes integrantes de la empresa donde se revisaba el estado del proyecto y se solucionaban las posibles dudas. Así mismo, una vez al mes aproximadamente, se realizó una reunión con el tutor del proyecto por parte de la Universidad. Todas estas reuniones se realizaron a través de la aplicación Microsoft Teams.

Inicialmente se realizó una reunión para definir la planificación. La empresa utiliza una metodología Agile, por ello, se troceó el proyecto en pequeñas tareas y subtareas que no durarían más de 1 - 2 semanas, las cuales han quedado reflejadas en la aplicación Trello, aplicación que utiliza la empresa para la gestión de los diferentes proyectos.

Para el alojamiento y control de versiones del código se utiliza el mismo servicio de alojamiento basado en web que utiliza la empresa, Bitbucket, en el cual se crearon diferentes ramas para el back end y front end. Cada actualización que se realizó del código se sometió a los tests ofrecidos por plugins instalados en el repositorio, encargados de controlar la calidad y seguridad del código, además se comprobó su calidad por otros compañeros de la empresa y se sometió a diferentes pruebas para confirmar su correcto funcionamiento y su correcta integración con la plataforma.

Para las tareas del back end se utilizó el framework LoopBack que funciona con TypeScript y Angular. Para almacenar los datos de la base de datos externa en local, se usó la base de datos de la empresa que utiliza un sistema de bases de datos NoSQL y orientado a documentos, como es MongoDB.

Por otra parte, para las tareas del front end se utilizó el framework Angular 8, con TypeScript, Pug y Sass. Además, se cuenta con la encargada de diseño de la empresa para realizar una interfaz visualmente correcta.

5 PLANIFICACIÓN

El proyecto se dividió en 19 semanas donde se dividieron cada uno de los objetivos en diferentes tareas y cada una de ellas contiene pequeñas subtareas, correspondientes a los diferentes subobjetivos, para el correcto desarrollo de las funcionalidades comentadas anteriormente. Como la empresa trabajaba con un marco de trabajo "Agile"

se adaptaron las diferentes subtareas para que tuvieran una duración aproximada de 1-2 semanas para posteriormente fusionarlas con la rama maestra de la versión de desarrolladores de la plataforma para poder comprobar su correcto funcionamiento y su correcta integración con el resto de la plataforma.

La Fig. 15 situada en el apartado 1 del apéndice muestra el diagrama de Gantt donde se puede observar el reparto de las diferentes tareas entre las múltiples semanas del proyecto.

6 DESARROLLO

A lo largo de este apartado, se expone el desarrollo de las tareas para los usuarios estándar, las tareas para los usuarios administradores y las tareas de testeo que se realizaron a lo largo de este proyecto.

Inicialmente se exponen las tareas para los usuarios estándar que empezó con la tarea de análisis inicial, seguidamente la de integrar Meteo con otros módulos, crear y configurar el mapa de Meteo y finalmente con la tarea de representar y editar los datos.

Seguidamente se exponen las tareas de los usuarios administradores que acabó y empezó con la tarea de crear y configurar el portal de administradores de meteo.

Finalmente se expone el conjunto de pruebas que se realizaron a la plataforma.

6.1. Tarea 0 - Análisis Inicial

En este apartado se expone el desarrollo de esta tarea donde se empezó identificando los casos de uso y la realización de los pertinentes diagramas. Seguidamente se realizaron todos los mock up y finalmente se estudió la API de PVGIS.

6.1.1. Subtarea 0.1. - Casos de uso

El primer paso fue la creación de los diagramas de casos de uso para poder definir correctamente los objetivos y los requisitos de los usuarios que han sido detectados como potenciales usuarios de este módulo, usuarios estándar y usuarios administradores.

El primer diagrama de caso de uso, Fig. 16, lo encontramos en el apartado 2 de los apéndices y es el del usuario estándar donde podremos observar las diferentes funcionalidades que tenían que poder realizar los usuarios una vez hubiese finalizado el proyecto.

La Fig. 1 muestra el diagrama de caso de uso del usuario administrador de Wattwin donde también se pueden observar las funcionalidades que tenían que poder realizar una vez finalizase el proyecto.

6.1.2. Subtarea 0.2. - Mock Up

El siguiente paso fue el desarrollo de los mock up o wireframe para planificar la estructura de los diferentes modelos y funcionalidades que se desarrollarían a lo largo del proyecto.

La Fig. 17 situada en el apartado 3 de los apéndices muestra uno de los 15 mock up que se han realizado a largo del proyecto y en este caso se muestra el diseño inicial de la página principal del módulo.

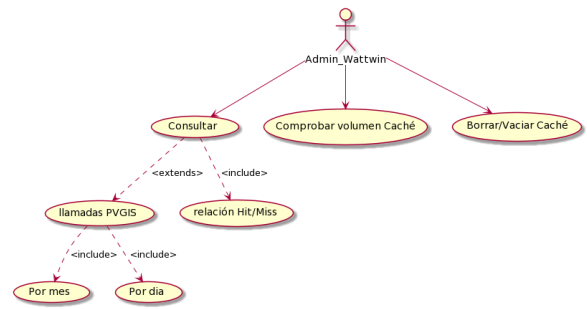


Fig. 1: Diagrama de Casos de uso para usuario administrador

6.1.3. Subtarea 0.3. - Estudio de la API PVGIS

Finalmente, se estudió la API PVGIS y se comprobó que ofrece cuatro herramientas que podrían ser interesantes para la realización de las funcionalidades de este módulo. Las cuatro herramientas son: radiación mensual, radiación diaria, radiación horaria y TMY (“Typical Meteorological Year”).

La Fig. 18 situada en el apartado 4 de los apéndices, muestra un diagrama donde aparecen los datos que se pueden obtener al utilizar cada una de esas herramientas.

La Fig. 2 muestra un diagrama donde aparece la tabla que se ha creado en la base de datos y los diferentes atributos que esta tiene.

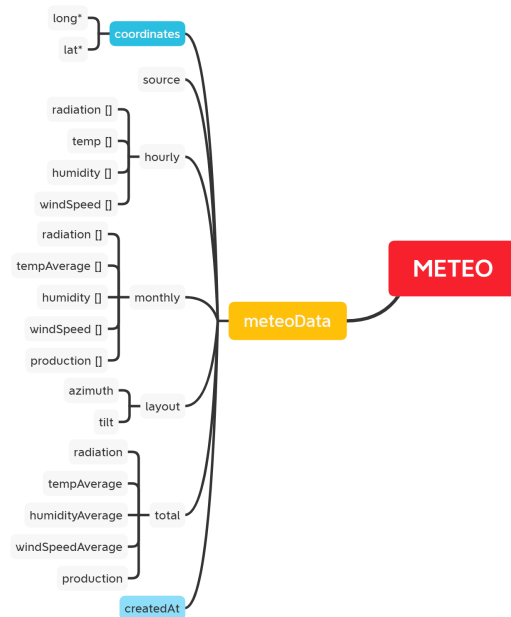


Fig. 2: Visualización añadir una dirección a caché

6.2. Tarea 1 - Integrar con otros módulos

Para sustituir el funcionamiento anterior de la plataforma, las tablas CTE, se decidió iniciar esta fase del proyecto realizando un algoritmo que se utiliza en toda la plataforma siempre que se solicite información sobre la radiación o la temperatura.

La lógica de este algoritmo inicia se recuperando de la

base de datos los datos del punto más cercano al solicitado. Si no existe ningún dato se realiza la petición a PVGIS para obtener datos del punto solicitado, se almacena en la base de datos y son presentados al usuario. Por otra parte, en el caso de que sí existan datos en la base de datos se comprueba si la ubicación de los datos recuperados está lo suficientemente cerca, en el caso de la plataforma se decidió que debía estar a menos de 10 kilómetros. Si los datos se encuentran a menos de 10 kilómetros son presentados al usuario ya que se consideró que en esa distancia la variación de la radiación o de la temperatura es asumible. Por otra parte, si los datos se encuentran a más de 10 kilómetros se realiza la petición a PVGIS para obtener datos del punto solicitado y se almacenan en la base de datos y, si todo se ha realizado correctamente, se presentan al usuario. En el caso de que se produzca algún error, se presentan al usuario los datos obtenidos del punto más cercano, datos que se han obtenido previamente.

Para la creación de este algoritmo se añadió un end point al modelo de Meteo donde se creó toda la lógica y se le dio formato a los datos.

La Fig. 3 expone un diagrama de estados del algoritmo comentado anteriormente.

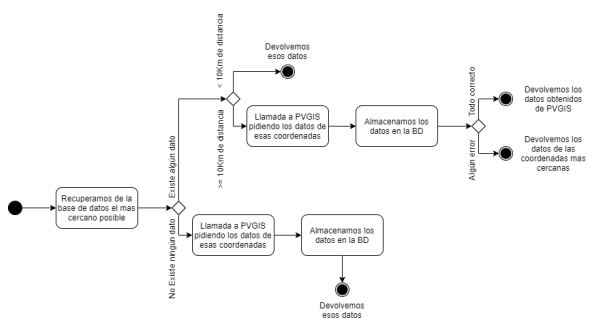


Fig. 3: Algoritmo de recuperación de los datos meteorológicos

A continuación se describen la subtarea de ajustar la radiación total solar a una planta fotovoltaica y la subtarea de adaptar la plataforma al nuevo formato de radiación.

6.2.1. Subtarea 1.1. - Ajustar radiación solar total a plata fotovoltaica

Gracias a las coordenadas, cuando se creaba una planta fotovoltaica dentro de la plataforma se ubicaba en una zona CTE. Una vez se ubicaba en una zona CTE, se extraían los datos anuales totales y horarios tanto fotovoltaicos como de la temperatura.

Actualmente, cuando se crea la planta fotovoltaica ya no se ubica dentro de ninguna zona CTE sino que esos datos se extraen de PVGIS, utilizando el algoritmo comentado anteriormente. Para ello, únicamente se tuvieron que crear instancias del modelo de Meteo para realizar llamadas al end point mencionado anteriormente.

Además, se habilitó una funcionalidad para que el usuario fuera capaz de recuperar la radiación total original de PVGIS si previamente había sido cambiado ese valor manualmente. Para la creación de esta funcionalidad fue necesario crear un end point en un modelo ya existente en la plataforma llamado “PowerPlant” y, una vez fue creado este end point, se utilizó la instancia del modelo para realizar

una llamada a ese end point desde un “button” creado en la parte de front end.

La Fig. 4 muestra el resultado del “button” para recuperar la radiación total, donde al pulsarlo se recuperarían los datos originales de PVGIS, utilizando el algoritmo comentado anteriormente para utilizar los datos almacenados en cache o realizar una petición a la base de datos externa.



Fig. 4: Botón de recuperación de la radiación total

6.2.2. Subtarea 1.2. - Adaptar nuevo formato de radiación

Al incorporar el sistema PVGIS a la plataforma, se tuvo que adaptar la forma en la que los métodos recuperaban los datos necesarios para realizar sus pertinentes funciones. Fue necesario realizar diferentes pruebas y asegurarse de su correcto funcionamiento, ya que no se debían permitir errores en este punto ya que podían afectar a toda la plataforma.

Otro de los cambios más relevantes fue establecer este nuevo sistema como el predeterminado en la creación de las nuevas plantas fotovoltaicas, con lo que también se tuvo que adaptar la lógica y las relaciones de los métodos que realizaban esta función.

La plataforma cuenta con unos “scripts” que calculan una estimación de ahorros en función de la ubicación del usuario. Estos cálculos también los realizaba en función de las coordenadas y de una zona CTE, con lo que también fue necesario adaptar los cálculos para que realicen esa estimación con los datos provenientes de PVGIS ya que posteriormente son los datos que se muestran al usuario.

Poco a poco se desea eliminar la lógica de las tablas CTE de la plataforma, el problema que se encontró fue que la plataforma se encuentra en funcionamiento y hay empresas que disponían de ofertas que no habían sido finalizadas y que habían sido creadas con los datos de las tablas CTE, lo que hacía que no fuera asumible que esos valores cambiaran de un día para otro al haber realizado estos cambios.

Por ese motivo se decidió mantener durante unos meses las tablas CTE dentro de la plataforma para que las plantas fotovoltaicas que fueron creadas antes de la incorporación de PVGIS a la plataforma realicen sus cálculos como lo hacían anteriormente y en el momento que se detecte que todas las ofertas activas utilizan el nuevo sistema se eliminará definitivamente las tablas CTE de la plataforma.

6.3. Tarea 2 - Crear y configurar mapa Meteo

El usuario debía de tener la posibilidad de establecer una dirección de tres maneras distintas indicando un punto en el mapa, indicando una dirección o indicando las coordenadas.

A continuación se explica la subtarea de consultar el mapa y seguidamente la subtarea de definir y consultar una ubicación.

6.3.1. Subtarea 2.1. - Consultar mapa

Para la creación del módulo lo primero que se realizó fue crear el componente principal, al que se le llamó “meteo-page-component”. La principal función de este componente es ser un contenedor de otros componentes, donde cada uno de ellos realiza pequeñas tareas.

El siguiente paso fue la creación de un componente donde situar el “heatmap”, se decidió llamarlo “meteo-heatmap-component”. Y finalmente se añadió este componente al componente principal del módulo.

Una vez ya se tenían estos dos componentes creados y en funcionamiento se decidió añadir el apartado de Meteo a la “sidebar” de la plataforma. Para ello se tuvieron que definir y crear las rutas y permisos necesarios para que los usuarios que posean esos permisos puedan acceder al módulo. Además se creó un script para añadir a la base de datos los permisos que se decidieron, lectura y lectura/escritura.

Cuando ya se podía acceder al módulo y que se pudo confirmar que el componente principal (meteo-page-component) y el subcomponente (meteo-heatmap-component) funcionaban correctamente fue el momento de crear el componente para el heatmap. Para la creación de ese componente únicamente se adaptó el heatmap que ofrecía Google, Heatmap Layer [11], con la intención de poder reutilizar este componente para futuros componentes de la plataforma, por ese motivo se le llamó “heatmap-component”.

Posteriormente se decidió no implementar un mapa de calor ya que el diseño que proporcionaba Google no fue el esperado por la empresa con lo que finalmente se retiraron los puntos en el mapa.

La Fig. 5 muestra la página principal del módulo donde se puede observar el mapa de la API de Google. Al definir una ubicación utilizando cualquiera de los métodos posibles aparecerá un “marker” en el mapa y se aplicará un efecto de zoom para dejar claro la ubicación que se está consultado.



Fig. 5: Página principal del módulo Meteo

6.3.2. Subtarea 2.2. - Definir y Consultar Ubicación.

La primera funcionalidad que se desarrolló fue la de definir la ubicación haciendo clic en el mapa. Únicamente se tuvo que configurar el mapa de Google para que cuando se realizara el evento de clic sobre él devolviera las coordenadas y se realizara una llamada al algoritmo de recuperación

de datos meteorológicos utilizando el end point que ha sido comentado anteriormente.

Seguidamente, se realizó la funcionalidad de definir la ubicación introduciendo las coordenadas. Para ello se desarrolló un Pop-up que al introducir las coordenadas y hacer clic en el botón de “buscar” realice también la llamada al algoritmo de recuperación de datos meteorológicos utilizando también el end point.

La Fig. 6 muestra el Pop-up para definir la ubicación utilizando las coordenadas. Al definir las coordenadas, si estas son correctas, se habilitará el botón de “buscar” para enviar esas coordenadas al algoritmo comentado anteriormente.

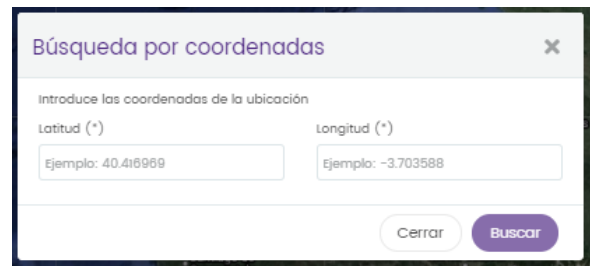


Fig. 6: Pop-up para definir las coordenadas

Finalmente se realizó la funcionalidad de definir la ubicación introduciendo la dirección, para ello se utilizó la API “places autocomplete”[12] ofrecida por Google. Para ello se creó un componente al que se le llamó “places-autocomplete” con la intención de poderlo reutilizar en diferentes partes de la plataforma. Este componente ofrece sugerencias en función de lo que está escribiendo el usuario y una vez se decide una dirección este devuelve información sobre esa dirección, entre las cuales se encuentran las coordenadas. Una vez obtenidas las coordenadas se realiza una llamada al algoritmo de recuperación de datos meteorológicos utilizando de nuevo el end point. Para tener un formato estándar de la dirección se ha realizando una función que formatea las direcciones antes de mostrarla en el resumen de los datos.

La Fig. 7 muestra el buscador de la API de Google adaptado al formato de la plataforma. Este buscador ofrece diferentes sugerencias en función de lo que se esta escribiendo y una vez se ha escogido la dirección se extraen las coordenadas para nuevamente enviárselas al algoritmo de recuperación de datos meteorológicos.

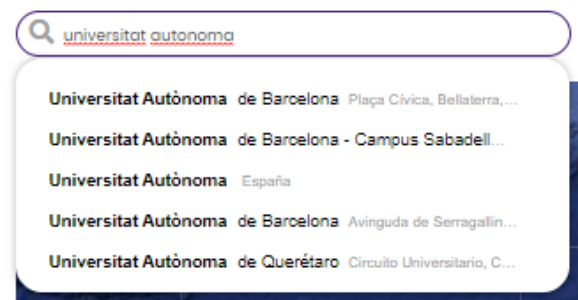


Fig. 7: Buscador de la API de Google

6.4. Tarea 3 - Representar y Editar los datos

Los datos deben de ser visibles por el usuario de una manera entendible, por este motivo se definió conjuntamente con la encargada del diseño de la empresa una disposición de los datos que fuera agradable para el usuario.

Además, entre los diferentes datos meteorológicos que se obtienen al definir una ubicación, se encuentra una estimación de producción eléctrica en función de una disposición óptima pero se decidió que estos datos debían de ser configurables por el usuario.

A continuación se explicarán las subtareas de Representar los datos y posteriormente la de Editar los datos representados.

6.4.1. Subtarea 3.1. Representar datos

Para la representación de los datos se decidió dividir estos en dos secciones.

La primera de ellas es un pequeño resumen donde se observarán los agregados anuales y las medias de los diferentes datos meteorológicos además de la dirección y las coordenadas. En esta misma sección se encuentra una pequeña subsección la cual se le llamó “Disposición óptima” donde encontramos la inclinación y orientación óptimas para obtener los datos que se observan en el resumen.

La Fig. 8 muestra la primera sección donde se pueden observar el resumen de los datos comentados. Estos datos son los que se obtendrían en una ubicación con la disposición óptima, disposición que, como comentaremos a lo largo del proyecto, se permite al usuario personalizar.



Fig. 8: Resumen de los datos

La segunda sección es un componente al que se le llamó “meteo-chart” y contiene un gráfico para cada uno de los datos (producción, radiación, temperatura, humedad y velocidad del viento) donde se puede observar un valor para cada hora durante un año al completo. Estos gráficos son creados utilizando la API NgxCharts [13].

Únicamente se tiene visible uno de los gráficos y permite al usuario cambiar de gráfico al hacer clic en un componente llamado “button-options”, componente que creó previamente un compañero.

La Fig. 9 muestra el gráfico de la producción mensual y se puede observar el componente “button-options” para cambiar de datos que mostrar. Este gráfico corresponde a la producción solar mensual en la ubicación pero, en los otros datos, se muestran los datos meteorológicos para cada hora del año.

6.4.2. Subtarea 3.2. Editar datos representados

En la primera sección que hemos comentado anteriormente encontrábamos la subsección “Disposición óptima”.

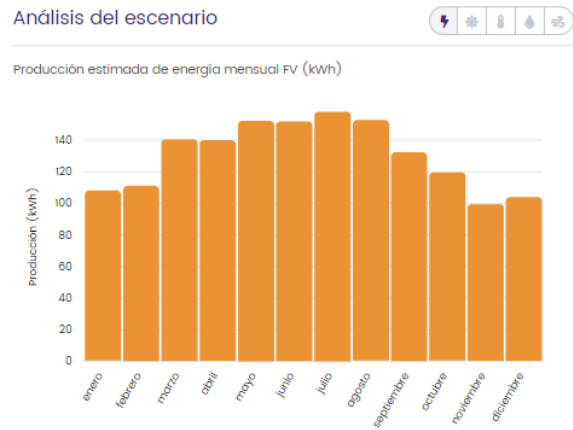


Fig. 9: Gráfico de barras representando la producción mensual

Aquí se encuentran los valores para la inclinación y orientación óptimas, valores que al pasar el ratón por encima podemos ver un efecto de “hover” que hará que aparezca un icono de un lápiz.

Este comportamiento lo proporciona un componente que ya existía previamente en la plataforma llamado “update-field” y se ha configurado para que al editar el valor realice una nueva petición a PVGIS para obtener la producción aproximada en ese punto con esa inclinación y esa orientación. Esta petición se hace en caliente y no se almacenan sus datos ya que son únicamente informativos para el usuario.

Una vez se obtiene respuesta de PVGIS se edita el valor del agregado mostrado en el resumen y se reconfigura el gráfico con los nuevos datos.

Finalmente comentar que la subsección después de ser editada pasaría a llamarse “Disposición” y al pasar por encima de ese título se puede observar como aparece el texto “Recuperar disposición óptima” que al hacer clic, se recuperan los datos previos a la edición de la inclinación y orientación.

La Fig. 10 muestra la edición de la inclinación y el botón de “Recuperar disposición óptima”. Al hacer clic en recuperar la disposición óptima se realizará una petición a PVGIS para poder recuperar los datos originales, por otra parte si se edita uno de los parámetros de la disposición se realizará una petición a PVGIS para obtener información meteorológica para esa disposición.

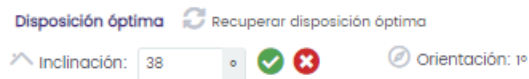


Fig. 10: Edición de la inclinación

6.5. Tarea 4 - Crear y Configurar portal de administración de Meteo

El desarrollo del módulo Meteo genera un gran tráfico de datos por la red. Con el fin de reducir este tráfico se decidió realizar el cacheado de los datos, ganando de esta manera también velocidad.

Al realizar el cacheado también se pierde fiabilidad ya que al almacenar los datos, si estos nunca se actualizan,

podrían ser erróneos con el paso del tiempo. Por este motivo se decidió realizar un portal de administración donde comprobar el número de direcciones almacenadas en caché y ofrecer a los administradores de Wattwin la posibilidad de eliminar las direcciones en un intervalo de tiempo con la finalidad de “refrescar” los datos almacenados en caché.

Finalmente se decidió también comprobar la calidad del cacheado realizando un gráfico para poder comprobar la relación entre aciertos en la caché y los fallos.

A continuación se explicarán las subtareas de comprobar el volumen de la cache, borrar la caché, consultar las llamadas a PVGIS, contar aciertos y fallos del cacheado y finalmente representar relación aciertos/fallos.

6.5.1. Subtarea 4.1. Comprobar volumen caché

Para este punto se utilizó una “number-card” de la API NgxCharts donde se le pasó la configuración básica (color, tamaño de letra, etc...), un texto para mostrar y el recuento de direcciones almacenadas.

Para saber el número de direcciones se creó un end point que únicamente cuenta cuantos documentos se encuentran almacenados en la tabla MeteoData, donde cada documento pertenecería a los datos meteorológicos de una ubicación.

La Fig. 11 muestra la “number-card” de la API NgxCharts que muestra el recuento de direcciones almacenadas.

9

Direcciones almacenadas

Fig. 11: Recuento de direcciones almacenadas

6.5.2. Subtarea 4.2. Borrar/Vaciar caché

Con la finalidad de poder definir el intervalo de fechas para eliminar los datos de la caché, se decidió crear un pop-up que apareciera al hacer clic el botón “Vaciar caché”. Este pop-up tiene dentro un componente llamado “range-field” donde se tuvo que especificar que eran dos fechas.

Al definir las dos fechas y hacer clic en el botón para confirmar, se hace una llamada a un end point que elimina todas las direcciones almacenadas entre esas fechas.

Finalmente se recargan los componentes necesarios para mostrar los datos actualizados.

La Fig. 12 muestra el pop-up para definir el intervalo de fechas que se desea eliminar. La fecha inicial del intervalo será la primera fecha que se encuentre entre los datos de la caché, al contrario que la fecha final que será la última fecha que se encuentre entre los datos de la cache.

6.5.3. Subtarea 4.3. Consultar/Contar llamadas PVGIS

Nuevamente utilizando los gráficos de la API NgxCharts se decidió añadir un gráfico para mostrar el número de llamadas a PVGIS, es decir, el número de veces que no se

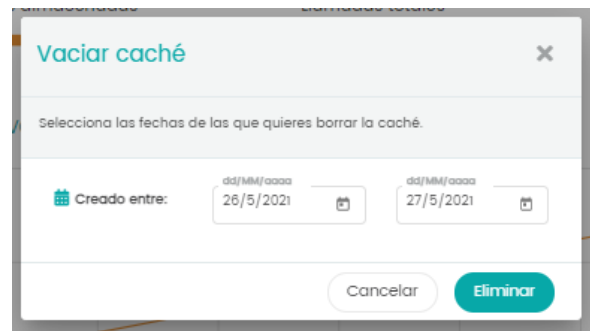


Fig. 12: Pop-up para vaciar la caché

encuentran los datos en caché. Estos datos se pueden visualizar tanto diariamente como mensualmente.

La Fig. 13 muestra el gráfico que representa el recuento de llamadas que se ha realizado por día.

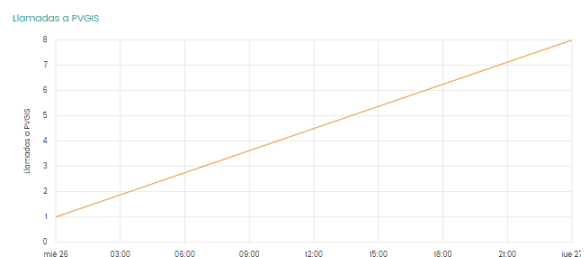


Fig. 13: Gráfico del recuento de llamadas por día

6.5.4. Subtarea 4.4. Contar aciertos y fallos de cacheado

Los fallos de caché eran fáciles de contar ya que se veían representados en el número total de direcciones almacenadas o en la gráfica de llamadas a PVGIS, pero en ese momento no se tenía la posibilidad de saber las veces que una petición había sido un acierto de la caché.

Por este motivo se añadió un “wrapper” a las peticiones del algoritmo de recuperación de datos meteorológicos para saber de donde se han obtenido los datos: de la API de PVGIS o directamente de caché.

6.5.5. Subtarea 4.5. Representar relación aciertos/fallos

Para la creación de esta petición se indexó ese “wrapper” a elastic search [14] y a través de dos peticiones se recuperaron el recuento de las peticiones diarias y mensuales para posteriormente mostrarlas en un gráfico de la API NgxCharts.

La Fig. 14 muestra el gráfico que representa la relación aciertos/fallos diarios donde la línea verde representaría los aciertos y la línea roja los fallos.

6.6. Tarea 5 - Definir y Ejecutar pruebas finales y Corregir errores

Para asegurar el correcto funcionamiento de este módulo se definieron diferentes casos imitando el posible uso diario de la plataforma y simulando diferentes situaciones que podrían ocurrir.

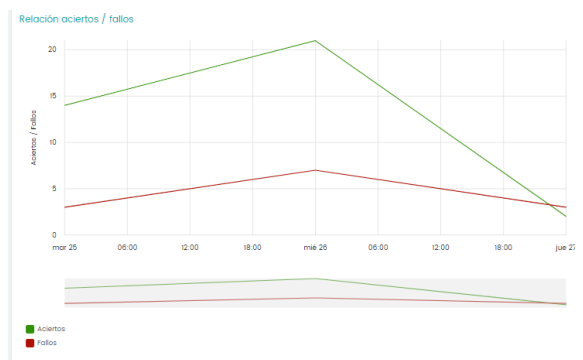


Fig. 14: Gráfico de la relación de aciertos/fallos por día

Posteriormente se ejecutaron esas pruebas y se corrigieron los errores encontrados para finalmente volver a empezar la tarea hasta el punto de encontrar el mayor número de errores posibles. A continuación se comentarán las sub-tareas definir pruebas, ejecutar pruebas y corregir pruebas.

6.6.1. Subtarea 5.1. Definir pruebas

Para las diferentes tareas que se han realizado durante el proyecto se pensaron diferentes escenarios para simular el comportamiento de un usuario utilizando el módulo.

El primer escenario que se pensó fue el caso que la API de PVGIS fallara. En este escenario siguiendo el algoritmo de recuperación de datos meteorológicos se debían devolver los datos del punto en caché más cercano posible.

El siguiente escenario que se pensó fue el caso que se solicitara un punto que estuviera a menos de 10 km de otro que había sido solicitado previamente. Siguiendo el algoritmo se deberían devolver esos datos.

Otro escenario que se pensó fue que se solicitara un punto que estuviera a más de 10 km de todos los puntos que habían sido solicitado previamente. Siguiendo nuevamente el algoritmo, se debería realizar una petición a PVGIS para devolver los datos de ese punto.

Finalmente el último de los escenarios que se pensó fue que se solicitara un punto que no cubriera la API de PVGIS con lo que nuevamente se deberían devolver los datos del punto mas cercano posible.

6.6.2. Subtarea 5.2. Ejecutar pruebas

Una vez se decidieron un conjunto de pruebas suficientes, se ejecutaron esas pruebas aisladamente para cada parte del módulo.

Gracias a esas pruebas se detectó que había un problema con la agregación realizada en MongoDB y no calculaba correctamente la distancia entre dos puntos; con lo que no estaba funcionando correctamente el cacheado de los datos, ya que siempre se solicitaban los datos a PVGIS.

6.6.3. Subtarea 5.3. Corregir errores

Finalmente, se corrigieron los diferentes errores que fueron detectados y se volvió a empezar la subtarea 5.1. para volver a iniciar el ciclo con la intención de encontrar más errores antes de que este módulo apareciera en producción.

7 RESULTADOS Y DISCUSIÓN

El proyecto ha acabado satisfactoriamente y dentro de los plazos establecidos, se han cumplido las diferentes tareas planteadas y se podría decir que han sido alcanzables para un ingeniero informático.

La integración de los diferentes módulos de la plataforma que utilizaban el mecanismo antiguo, las tablas CTE, ha sido todo un éxito y ha hecho que se realicen menos cálculos en el back end.

En un principio se tenía pensado crear un mapa de calor para visualizar la radiación solar en un punto del mapa. Después de su desarrollo se decidió eliminarlo, ya que el resultado que ofrecía la API de Google Maps no era el deseado por la empresa y finalmente se decidió dejar únicamente el mapa permitiendo definir la ubicación de las tres maneras posibles.

Una vez definida la ubicación, la representación de los datos ha quedado de manera que sea fácilmente entendible por el usuario a la vez que cómoda para su lectura, al igual que la personalización de la disposición de la planta FV.

Otro de los objetivos que en un principio se tenía en mente desarrollar fue el de exportación de datos. Finalmente se decidió retirar de este proyecto ya que lo alargaría en exceso y se consideró que las otras tareas tenían más importancia y se podía desarrollar en un proyecto futuro.

La creación de este módulo generaba un gran volumen de datos para almacenar en la base de datos con tal de realizar un cacheado de la información, con la finalidad de no tener que realizar constantemente peticiones a la API de PVGIS. Para poder comprobar tanto el volumen de datos, como si está funcionando correctamente el cacheado de los datos y purgar un intervalo de datos de la base de datos, se decidió crear el portal de administración donde poder realizar esas funciones.

Si comparamos el resultado obtenido con la propia API de PVGIS veremos que en este módulo tanto la estética como la disposición de los datos facilitan mucho su entendimiento y lo hacen mucho más eficiente para conseguir datos meteorológicos juntando este servicio a los diferentes servicios que ofrece la plataforma.

8 CONCLUSIONES

La empresa SIIE ofrece una solución digital para el diseño, venta y gestión de instalaciones fotovoltaicas a través de la plataforma Wattwin. Inicialmente, la plataforma no ofrecía información fiable sobre la producción y radiación solar, ya que utilizaban un mecanismo a través de unas tablas CTE, tablas que no se actualizaban dinámicamente.

Se decidió desarrollar un proyecto donde a través de la API PVGIS ofrecer a sus clientes información meteorológica fiable. De este proyecto surgió Meteo, un módulo de la plataforma Wattwin que ofrece información meteorológica sobre un punto permitiendo personalizar la disposición de la planta fotovoltaica y obtener la producción en relación a esa disposición.

La integración con otros módulos, la creación del mapa, la representación y edición de los datos y la creación del portal de administración ha sido todo un éxito y en la siguiente “release” será subido a producción.

Gracias al portal de administración se ha podido comprobar que cuanto más pasa el tiempo más eficiente está siendo el cacheado, ya que poco a poco se tienen los datos almacenados en la base de datos de la plataforma.

Próximamente se añadirá al módulo una opción para la exportación de los datos en diferentes formatos, funcionalidad que se tenía pensada implementar desde un primer momento. Otra funcionalidad que se quiere desarrollar en un futuro es exportar los datos de módulo a una oferta para el cliente, con la finalidad de optimizar el tiempo.

9 FUENTES DE INFORMACIÓN

Para la realización del proyecto se consultaron las diferentes páginas oficiales del software, ya que contienen ejemplos e información suficiente. Además, se contó con la propia documentación y software de la empresa que se encontraba disponible para obtener información útil. Finalmente, también se dispuso de ejemplos encontrados en la red y en la propia plataforma.

AGRADECIMIENTOS

Me gustaría agradecer al equipo de SIIE por darme la oportunidad de formar parte de su equipo, desarrollar este proyecto y ofrecerme su ayuda siempre que ha sido necesario. También me gustaría agradecer al tutor de este TFG, Ramón Martí, por los consejos y las correcciones de este documento para ayudarme a confeccionar un informe digno para un TFG de ingeniería.

REFERENCIAS

- [1] SIIE. (2021, Marzo 01). Consultancy and BPO services for the solar PV industry. [En línea]. Disponible: <http://sii-e.com/>
- [2] Wattwin. (2021, Marzo 03). La solución digital para el diseño, venta y gestión de instalaciones fotovoltaicas. [En línea]. Disponible: <https://wattwin.com/>
- [3] PVGIS. (2021, Marzo 03). Photovoltaic Geographical Information System (PVGIS). [En línea]. Disponible: <https://ec.europa.eu/jrc/en/pvgis/>
- [4] MongoDB. (2021, Marzo 03). The MongoDB 4.2 Manual. [En línea]. Disponible: <https://docs.mongodb.com/v4.2/>
- [5] Angular 8. (2021, Marzo 03). Introduction to the Angular Docs. [En línea]. Disponible: <https://angular.io/docs/>
- [6] LoopBack 3. (2021, Marzo 03) A highly extensible Node.js and TypeScript framework for building APIs and microservices. [En línea]. Disponible: <https://loopback.io/doc/en/lb3/>
- [7] PUG 3 . (2021, Marzo 03). Página oficial. [En línea]. Disponible: <https://pugjs.org/>
- [8] Sass 3. (2021, Marzo 03). Documentation. [En línea]. Disponible: <https://sass-lang.com/documentation>
- [9] TypeScript. (2021, Marzo 03). TypeScript Documentation [En línea]. Disponible: <https://www.typescriptlang.org/docs/>
- [10] Bitbucket. (2021, Marzo 12). Bitbucket, página oficial. [En línea]. Disponible: <https://bitbucket.org/>
- [11] Heatmap Layer. (2021, Abril 21). Google Maps Platform. Heatmap Layer. [En línea]. Disponible: <https://developers.google.com/maps/documentation/javascript/heatmap>
- [12] Places Autocomplete. (2021, Mayo 18). Google Maps Platform. Places Autocomplete. [En línea]. Disponible: <https://developers.google.com/maps/documentation/places/web-service/autocomplete>
- [13] NgxCharts. (2021, Mayo 18). NGX-CHARTS 18.0.1ANGULAR CHART FRAMEWORK. [En línea]. Disponible: <https://swimlane.github.io/ngx-charts/>
- [14] Elastic Search. (2021, Mayo 19). Elastic Search. [En línea]. Disponible: <https://www.elastic.co/es/>

APÉNDICE

A.1. Gantt

En la Fig. 15 se puede observar el diagrama de Gantt donde se puede ver el reparto de las diferentes tareas entre las diferentes semanas.

A.2. Casos de uso estándar

En la Fig. 16 muestra el diagrama de casos de uso del usuario estándar que será el usuario que utilizará las funcionalidades del módulo como definir una ubicación o consultar los datos.

A.3. Mock up

En la Fig. 17 se puede observar el diseño inicial para la página principal del módulo Meteo donde se puede apreciar el mapa, el buscador de Google y el botón para abrir el pop-up para definir unas coordenadas.

A.4. PVGIS

En la Fig. 18 muestra los diferentes intervalos y los diferentes datos que la API PVGIS permite solicitar. Se puede observar como algunos de esos datos están resaltados en color rojo/salmón ya que esos datos son los que utiliza este módulo.

| Nombre de la tarea | Subtareas | Semana 1 | Semana 2 | Semana 3 | Semana 4 | Semana 5 | Semana 6 | Semana 7 | Semana 8 | Semana 9 | Semana 10 | Semana 11 | Semana 12 | Semana 13 | Semana 14 | Semana 15 | Semana 16 | Semana 17 | Semana 18 | Semana 19 | |
|--|--|------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|--|
| | | Identificar Objetivos | | ■ | | | | | | | | | | | | | | | | | |
| Identificar Requisitos | | | | | | | | | | | | | | | | | | | | | |
| Organizar Tareas y Subtareas | | | ■ | | | | | | | | | | | | | | | | | | |
| Tarea 0 - Análisis Inicial | | | | | | | | | | | | | | | | | | | | | |
| | 0.1. Casos de Uso | ■ | | | | | | | | | | | | | | | | | | | |
| | 0.2. Mock Up | | | ■ | | | | | | | | | | | | | | | | | |
| | 0.3. Estudio API PVGIS | | | | ■ | | | | | | | | | | | | | | | | |
| Tarea 1 - Integrar con otros módulos | | | | | | | | | | | | | | | | | | | | | |
| | 1.1. Ajustar radiación solar total a planta FV | | | | | ■ | ■ | | | | | | | | | | | | | | |
| | 1.2. Adaptar nuevo formato de radiación / Eliminar antiguo | | | | | | | ■ | | | | | | | | | | | | | |
| Tarea 2 - Crear y Configurar Mapa Meteo | | | | | | | | | | | | | | | | | | | | | |
| | 2.1. Consultar mapa | | | | | | | | | ■ | ■ | | | | | | | | | | |
| | 2.2. Definir y Consultar Ubicación | | | | | | | | | | | | | | | | | | | | |
| Tarea 3 - Representar y Editar datos | | | | | | | | | | | | | | | | | | | | | |
| | 3.1. Representar datos | | | | | | | | | | | | ■ | | | | | | | | |
| | 3.2. Editar datos representados | | | | | | | | | | | | | | | | | | | | |
| Tarea 4 - Crear y Configurar portal de administración de Meteo | | | | | | | | | | | | | | | | | | | | | |
| | 4.1. Comprobar volumen caché | | | | | | | | | | | | | | | | | | | | |
| | 4.2. Borrar/Vaciar Caché | | | | | | | | | | | | | | | | | | | | |
| | 4.3. Consultar/Contar llamadas PVGIS | | | | | | | | | | | | | | | | | | | | |
| | 4.4. Contar aciertos y fallos de cacheado | | | | | | | | | | | | | | | | | | | | |
| | 4.5. Representar relación aciertos/fallos | | | | | | | | | | | | | | | | | | | | |
| Tarea 5 - Definir y Ejecutar pruebas finales y Corregir errores | | | | | | | | | | | | | | | | | | | | | |
| | 5.1. Definir pruebas | | | | | | | | | | | | | | | | | | | | |
| | 5.2. Ejecutar pruebas | | | | | | | | | | | | | | | | | | | | |
| | 5.3. Corregir errores | | | | | | | | | | | | | | | | | | | | |
| Tarea 6 - Finalizar documentación | | | | | | | | | | | | | | | | | | | | | |
| | 6.1. Finalizar paper | | | | | | | | | | | | | | | | | | | | |
| | 6.2. Finalizar presentación | | | | | | | | | | | | | | | | | | | | |
| | 6.3. Realizar presentación | | | | | | | | | | | | | | | | | | | | |

Fig. 15: Diagrama de Gantt del proyecto Meteo

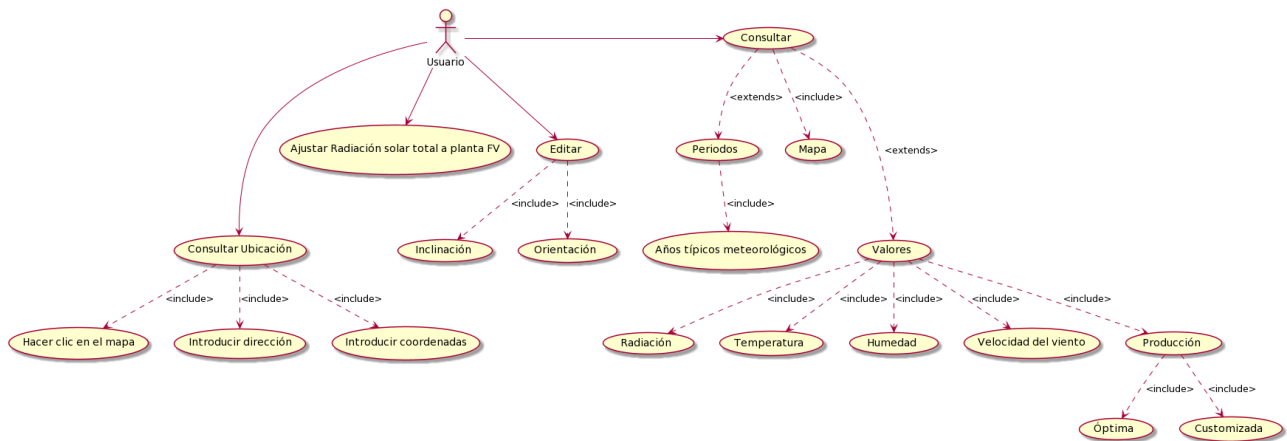


Fig. 16: Diagrama de Casos de uso del usuario estándar

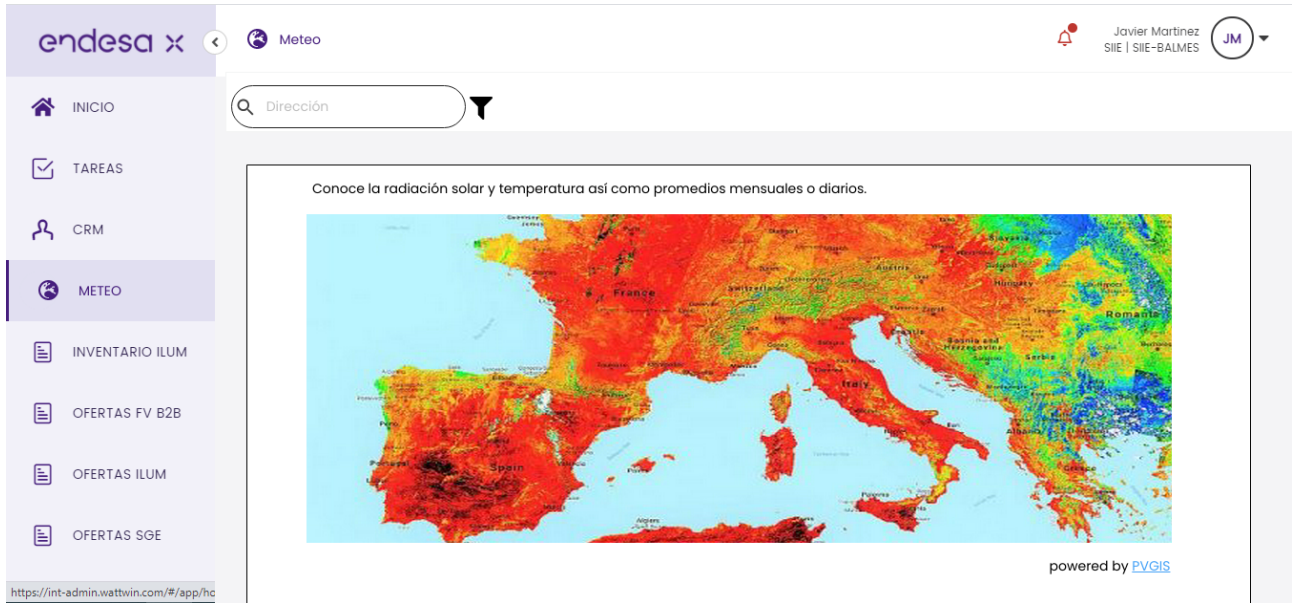


Fig. 17: Mock up entrada al módulo Meteo

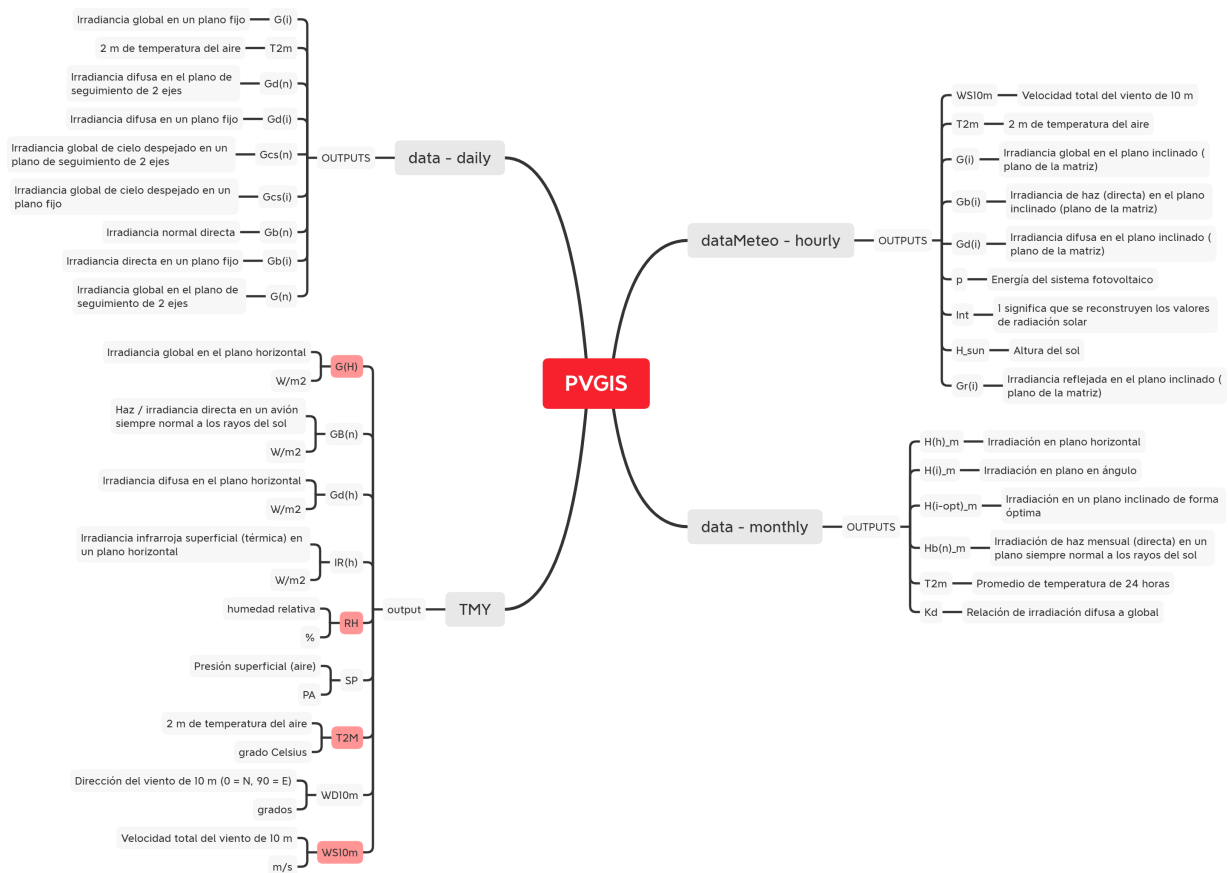


Fig. 18: Estructura base de datos PVGIS