



---

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

García Benjumea, Hugo; Figuerola Torrell, Adrià, dir. Diseñador fotovoltaico 3D en plataforma Wattwin. 2024. (Enginyeria Informàtica)

---

This version is available at <https://ddd.uab.cat/record/298968>

under the terms of the  license

# Diseñador fotovoltaico 3D en plataforma Wattwin

Hugo García Benjumea

30 de junio de 2024

**Resumen—** En el momento actual de cambio hacia fuentes de energía renovables, entre ellas, la energía solar, este proyecto se enfoca en mejorar el diseñador fotovoltaico existente en la plataforma Wattwin con la incorporación de funcionalidades avanzadas para considerar sombras generadas por obstáculos. Asimismo, se desarrolla un visor 3D para simular la incidencia de la luz solar y las sombras en tiempo real, además de integrar la Solar API de Google para enriquecer los datos. La metodología Scrum Agile guía el desarrollo, dividiendo el proyecto en fases con objetivos específicos. Las mejoras implementadas, que incluyen la gestión de obstáculos, el cálculo de sombras preciso y la visualización en 3D, proveen al usuario de una herramienta actualizada, avanzada y robusta para optimizar el diseño de instalaciones fotovoltaicas.

**Palabras clave—** Gestión de obstáculos, cálculo de sombras, visualización 3D, Solar API, metodología Scrum Agile, diseño de instalaciones fotovoltaicas

## 1 INTRODUCCIÓN

**T**RANSICIONAR hacia fuentes de energía renovables es, actualmente, más crucial que nunca, y dentro de este contexto la energía solar destaca como una solución sostenible y cada vez más accesible [1][2]. La implementación eficaz de sistemas fotovoltaicos depende en gran medida de una planificación y diseño detallados que maximicen su rendimiento energético. En este escenario, la tecnología juega un papel fundamental, facilitando herramientas avanzadas que permiten optimizar el diseño y la producción energética de instalaciones fotovoltaicas.

Este proyecto se centra en mejorar el diseñador fotovoltaico ya existente en la plataforma Wattwin, que actualmente utiliza Google Maps para permitir a los usuarios diseñar instalaciones solares mediante la selección de áreas específicas [3]. Sin embargo, una limitación significativa de la herramienta actual es la incapacidad para considerar la influencia de elementos que pueden generar sombras sobre las áreas seleccionadas para la instalación de paneles solares. La sombra proyectada puede reducir drásticamente la eficiencia de los mismos, lo que hace imprescindible su consideración durante la fase de diseño.

Por lo tanto, el objetivo principal de este trabajo es abor-

dar esta limitación integrando funcionalidades avanzadas que permitan a los usuarios indicar la presencia de obstáculos y su altura correspondiente, consiguiendo así, gracias al procesamiento de estos datos, calcular la sombra generada por dichos obstáculos a lo largo del año. También permitir simular el impacto de estas sombras en los paneles solares en tiempo real de forma totalmente visual. Para lograr esto, se explorará la implementación de un visor 3D que permita simular la incidencia de la luz solar sobre los paneles solares en cualquier día y hora del año, utilizando tecnologías de renderizado 3D para navegadores. Además, se integrará Solar API de Google para enriquecer el cálculo y visualización de las sombras, además de aumentar la cantidad de información que se muestra al usuario en tiempo real [4].

Este desarrollo no solo representa un avance técnico en el diseño de instalaciones fotovoltaicas, sino que también contribuye al esfuerzo global por optimizar el aprovechamiento de la energía solar, una piedra angular en la transición hacia un futuro energético sostenible.

La visión general de este artículo es la siguiente. En primer lugar, en la sección 2 se definen los objetivos que se pretenden alcanzar. En segundo lugar, en la sección 3 se describe la metodología empleada, incluyendo también las herramientas utilizadas para llevarla a cabo en este proyecto. En tercer lugar, en la sección 4 se expone el estado del arte en los diferentes aspectos del trabajo. En cuarto lugar, en la sección 5 se propone la planificación para la realización de las diversas funcionalidades a crear. En quinto lugar, en la sección 6 se describe la implementación de la gestión de obstáculos en el diseñador actual. En sexto lugar, en la

- E-mail de contacto: hugo.garciabe@autonoma.cat
- Mención realizada: Tecnologías de la Información
- Trabajo tutorado por: Adrià Figuerola Torrell
- Curso 2023/24



sección 7 se explican las funciones y los diferentes algoritmos utilizados para realizar el cálculo anual del porcentaje de sombras sobre cada módulo solar. En séptimo lugar, en la sección 8 se desarrolla la creación de un visor 3D. En octavo lugar, en la sección 9, se presentan los datos y funcionalidades de Solar API usados en la implementación del visor 3D. Finalmente, en la sección 10 se exponen las conclusiones de este proyecto, incluyendo futuras mejoras y funcionalidades a desarrollar.

## 2 OBJETIVOS

El objetivo principal es mejorar y optimizar el proceso de diseño de instalaciones fotovoltaicas mediante el desarrollo e integración de tecnologías avanzadas de visualización y simulación en 3D. Este gran objetivo se desglosa en varios objetivos específicos, detallados a continuación:

- **Considerar elementos que generan sombras en el diseño de instalaciones fotovoltaicas.** Desarrollar una funcionalidad que permita a los usuarios indicar la presencia y altura de obstáculos que puedan generar sombras sobre las áreas destinadas a la instalación de paneles solares, tales como edificios, árboles u otros elementos colindantes. Esto implica la habilidad de mapear estos obstáculos en la plataforma actual de diseño basada en Google Maps.
- **Implementar un sistema de cálculo de sombras en tiempo real.** Utilizar la información sobre obstáculos para calcular, en tiempo real, el impacto de las sombras generadas por estos a lo largo del año sobre los paneles solares. Esto incluye el desarrollo de algoritmos que simulen la trayectoria solar y evalúen la incidencia de sombras en cada momento, ofreciendo a los usuarios datos precisos sobre el porcentaje de sombra en cada panel mediante *web-sockets*.
- **Desarrollar un visor 3D para la simulación de instalaciones fotovoltaicas.** Crear un visor 3D que permita visualizar en detalle y con profundidad la instalación fotovoltaica diseñada, incluyendo la simulación de la incidencia del sol y las sombras generadas por los obstáculos identificados. Esto requerirá la selección e implementación de tecnologías web adecuadas para el renderizado 3D en navegadores.
- **Incorporar datos y funcionalidades de la Solar API de Google.** Investigar cómo integrar la Solar API de Google en el desarrollo del visor 3D y/o en el cálculo de sombras, con el fin de enriquecer la herramienta de diseño con datos precisos sobre la radiación solar y otros factores relevantes para el diseño fotovoltaico.

Estos objetivos están orientados a superar las limitaciones actuales de las herramientas de diseño fotovoltaico en Wattwin, proporcionando a los usuarios una plataforma más precisa, interactiva y eficiente para el diseño de instalaciones solares, lo que a su vez contribuirá significativamente a optimizar el aprovechamiento de la energía solar.

## 3 METODOLOGÍA

Para el desarrollo de este proyecto se usará una metodología Scrum Agile, un marco de trabajo iterativo e incremental que facilita la gestión de proyectos complejos mediante la división del trabajo en ciclos cortos y manejables, conocidos como sprints [5]. Esta metodología es particularmente adecuada para proyectos de desarrollo de software que requieren flexibilidad y adaptabilidad ante requisitos cambiantes o emergentes. A continuación, se detalla cómo se aplicará esta metodología en el contexto de este proyecto.

Inicialmente, se llevará a cabo una fase de planificación en la que se definirán los objetivos generales del proyecto, se identificarán las funcionalidades y requisitos esenciales, y se creará el *Product Backlog*. Este es un listado dinámico que contiene todas las tareas, funcionalidades y requisitos técnicos necesarios para desarrollar el proyecto.

Al inicio de cada sprint, que tendrá una duración de 3 semanas, se realizará una reunión de planificación en la que el equipo de desarrollo, junto con el *Product Owner*, seleccionará los elementos del *Product Backlog* que se trabajarán durante ese sprint. Esta selección se basará en las prioridades establecidas y en la capacidad del equipo.

Durante cada sprint, el equipo llevará a cabo reuniones diarias de no más de 15 minutos. En estas reuniones, cada miembro del equipo informará sobre lo que ha trabajado el día anterior, lo que planea trabajar ese día y cualquier impedimento que pueda afectar su trabajo. Esto promueve la comunicación continua y permite identificar y resolver problemas rápidamente.

Cada semana, el equipo realizará una reunión pre-review para evaluar el progreso hacia los objetivos del sprint. Esta reunión servirá para ajustar estrategias, redefinir prioridades y asegurar que el proyecto se mantiene en el camino correcto, a la vez que tratar de resolver posibles dudas y/o problemas que hayan surgido durante la semana.

Por último, al final de cada sprint, de 3 semanas de duración, se llevará a cabo una reunión de revisión en la que el equipo presentará lo que ha sido completado durante el sprint. Este es un momento para demostrar nuevas funcionalidades, recibir feedback y ajustar el *Product Backlog* si es necesario. En este punto también se realizará una retrospectiva para reflexionar sobre el proceso del sprint, identificar qué funcionó bien, qué se puede mejorar y cómo se pueden implementar esos cambios en los próximos sprints.

### 3.1. Herramientas

Para apoyar y facilitar la implementación de la metodología Scrum Agile, se utilizarán diversas herramientas.

- **Tableros Kanban:** Los tableros Kanban son herramientas visuales que permiten a los equipos gestionar y visualizar el flujo de trabajo durante los sprints de una forma intuitiva y sencilla de entender [6][7]. Estos tableros suelen estar divididos en columnas que representan diferentes estados de las tareas (por ejemplo, en inglés, "To do", "Work in progress", "QA", "Done"), y las tareas se representan como tarjetas que se mueven de una columna a otra a medida que avanzan en su proceso de desarrollo.

- **Software de gestión de proyectos:** Se utilizarán diversas herramientas de gestión de proyectos para facilitar la organización, seguimiento y comunicación entre los diferentes miembros del equipo. Algunas de las herramientas usadas, del conjunto de productos de Atlassian, incluyen Sourcetree, Jira y Confluence [8]. Sourcetree es una herramienta de gestión que proporciona una interfaz gráfica intuitiva para trabajar con repositorios Git, permitiendo a los desarrolladores visualizar el historial de cambios, gestionar ramas y realizar fusiones, entre otras funcionalidades, de manera fácil y eficiente [9][10]. Jira es una plataforma de gestión de proyectos que permite a los equipos planificar, rastrear y gestionar el trabajo en un entorno colaborativo, donde los equipos pueden crear y asignar tareas, establecer fechas límite, realizar seguimiento del progreso y colaborar en tiempo real [11]. Por su parte, Confluence es un entorno colaborativo en el que los equipos pueden crear, compartir y colaborar en documentos, diagramas y otros contenidos, facilitando la documentación detallada del proyecto, la organización de reuniones y la comunicación entre los miembros del equipo [12].

Estas herramientas proporcionan un conjunto completo de funcionalidades para la gestión eficiente de proyectos ágiles como este, facilitando la colaboración y garantizando un seguimiento preciso del trabajo realizado.

## 4 ESTADO DEL ARTE

Para abordar los desafíos relacionados con la interacción del entorno en la producción de energía solar fotovoltaica, es crucial aprovechar tanto herramientas consolidadas como nuevas tecnologías emergentes. Wattwin, una plataforma integral para el diseño de instalaciones fotovoltaicas, ha demostrado ser una herramienta valiosa al permitir a los usuarios definir diversos parámetros, como orientación, inclinación y cantidad de módulos, para realizar estimaciones de producción. Sin embargo, una limitación importante de estas herramientas es su capacidad para considerar las sombras proyectadas por objetos circundantes.

Una aproximación común para abordar este problema es el uso de modelos de elevación digital (DEM) junto con algoritmos de trazado de rayos. Plataformas como PVsyst y Helioscope utilizan este enfoque para proporcionar análisis detallados de sombreado y evaluaciones de rendimiento [13][14]. Este método permite modelar la topografía del terreno y la ubicación de obstáculos, lo que resulta en una estimación más precisa de la pérdida de energía debido a sombras. Estos modelos y algoritmos han sido fundamentales para mejorar la precisión en el diseño de sistemas fotovoltaicos, destacándose por su capacidad de simular condiciones reales de sombra a lo largo del año.

Además de calcular y optimizar la producción, la visualización en 3D de las instalaciones es una herramienta invaluable. Esto permite a los usuarios ver una representación realista del diseño y realizar simulaciones. Para lograr esto en entornos web, se han desarrollado tecnologías de renderizado avanzadas, como Three.js y Babylon.js [15][16].

Babylon.js y Three.js son dos de las bibliotecas más populares para el renderizado 3D en navegadores. Babylon.js destaca por su robustez y su facilidad de uso en la creación

de experiencias interactivas y realistas en navegadores web, además de por su soporte para una amplia gama de características avanzadas, como la física y el procesamiento de escenas complejas. Por otro lado, Three.js es conocida por su simplicidad y su extensa documentación, lo que facilita su adopción por parte de desarrolladores. Sin embargo, para este proyecto se ha optado por Babylon.js debido a su capacidad para manejar escenas complejas y su integración más fluida con otras tecnologías web utilizadas en la plataforma.

Para el cálculo de sombras se han considerado varias técnicas: algoritmo de trazado de rayos y modelos de elevación digital (DEM).

1. **Algoritmo de trazado de rayos:** Este método simula la propagación de la luz solar y evalúa las áreas de sombra mediante la detección de intersecciones entre rayos y obstáculos. Es altamente preciso y adecuado para modelar sombras dinámicas a lo largo del día y el año. El trazado de rayos se basa en enviar rayos desde la fuente de luz (sol) y detectar las intersecciones en la escena. Si un rayo intersecta con un obstáculo antes de llegar al plano de los paneles solares, se considera que el punto de impacto está en sombra. Esta técnica proporciona una simulación realista de las sombras pero puede ser computacionalmente intensiva, especialmente en escenas con muchos obstáculos [17][18].
2. **Modelos de elevación digital (DEM):** Utilizan datos geoespaciales para representar la topografía del terreno y calcular sombras estáticas. Este método, útil para grandes áreas, proporciona una representación precisa de la elevación del terreno y los objetos sobre él. Los DEMs calculan sombras basándose en la elevación y la posición del sol en un momento dado, produciendo mapas de sombra estática. Aunque es eficiente para grandes áreas y útil para un análisis inicial, puede no ser tan preciso para análisis detallados en instalaciones específicas debido a su naturaleza estática y la falta de consideración de variaciones temporales [19].

La elección del algoritmo de trazado de rayos como técnica para la realización de este proyecto se basa en su precisión y capacidad para simular la incidencia de sombras en tiempo real, lo cual es esencial para el diseño óptimo de instalaciones fotovoltaicas.

La importancia de este proyecto radica en su capacidad para proporcionar a los usuarios una herramienta más precisa y detallada para el diseño de instalaciones solares, abordando de manera efectiva los desafíos relacionados con las sombras y el entorno circundante. Al integrar visualización 3D y cálculo preciso de sombras, se busca optimizar la ubicación y la orientación de los paneles solares, maximizando así la eficiencia y la producción de energía. Este avance no solo representa una mejora técnica significativa sino que también contribuye al esfuerzo global por aumentar la adopción de energías renovables y reducir la dependencia de fuentes de energía no sostenibles.

## 5 PLANIFICACIÓN

La planificación del proyecto se divide en varias fases, cada una enfocada en objetivos específicos dentro del marco de la metodología Scrum Agile, adaptada a las necesidades

de este proyecto. Se organizarán sprints de tres semanas, seguidos de evaluaciones y ajustes necesarios para el sprint siguiente. Se puede ver el diagrama de Gantt correspondiente a la planificación siguiente en la figura 12.

- **Semana 1-2:** En la primera fase se recopilarán los requisitos del proyecto y se establecerán las bases del sistema de cálculo de sombras, incluyendo tanto la gestión de los obstáculos como también el propio algoritmo de cálculo, y del diseño del visor 3D. Se dedicará a la investigación de la tecnología de visualización y simulación en 3D, Babylon.js, y a la exploración de la Solar API de Google para su posible integración. Se revisarán los hallazgos y se ajustará la planificación del proyecto en función de la viabilidad técnica y los recursos disponibles.
- **Semana 3-6:** Esta fase marca el inicio del desarrollo, con un enfoque especial en la integración de funcionalidades para el manejo de obstáculos que generan sombras y el inicio del desarrollo del cálculo de sombras. Al final de esta fase se lanzará la funcionalidad de gestión de obstáculos y una primera versión del cálculo de sombras. Se demostrará el prototipo para recopilar feedback, sobre el cual se realizarán los ajustes necesarios para incorporar funcionalidades adicionales.
- **Semana 7-10:** Continuando con el desarrollo, esta fase estará centrada en la realización del cálculo de sombras, incluyendo la comunicación en tiempo real de estos valores calculados. Se llevarán a cabo pruebas intensivas de las nuevas funcionalidades y se recopilará feedback para la corrección de errores y la optimización del rendimiento y la usabilidad. En esta fase también se integrará la funcionalidad de auto-cálculo de sombras al modificar el diseño.
- **Semana 11-13:** El objetivo será la creación del Visor 3D para permitir al usuario visualizar el diseño realizado en un entorno tridimensional. También se integrará la Solar API de Google para aumentar la precisión del diseño, lo que permitirá mejorar ajustes para maximizar la eficiencia. Se llevarán a cabo pruebas para comprobar que la instalación realizada en 2D se plasma correctamente en el entorno 3D y se integra Solar API.
- **Semana 14:** En esta fase, el enfoque estará en la integración de los diferentes componentes desarrollados, asegurando que el sistema de cálculo de sombras, el visor 3D y la integración con la Solar API de Google funcionen de manera cohesiva con el resto de la plataforma. Se evaluará la funcionalidad completa en un entorno similar al de producción para garantizar que todos los componentes interactúan correctamente.
- **Semana 15:** Esta semana se dedicará a la finalización del desarrollo, con un enfoque en el refinamiento de detalles, optimización del rendimiento y preparación de la herramienta para su lanzamiento.

Como se ha comentado en la sección 3 cada fase incluirá diferentes reuniones. Este enfoque garantiza que el proyecto se mantenga en el camino correcto y permite la adaptación a cualquier cambio o desafío que pueda surgir durante el desarrollo.

Durante las primeras dos semanas, se llevaron a cabo investigaciones y planificaciones que condujeron a ajustes significativos en la planificación del proyecto. En particular, se decidió dividir el desarrollo en más fases para mejorar la gestión del proyecto y garantizar una implementación más fluida de las funcionalidades. Esta reorganización implicó la creación de nuevas etapas, concretamente la fase dedicada exclusivamente a la gestión de obstáculos y la fase adicional para el desarrollo del visor 3D. Estos cambios se realizaron para garantizar la viabilidad técnica y mejorar la eficiencia y calidad del proceso de desarrollo, asegurando que cada aspecto del proyecto se aborde de manera adecuada y en el tiempo previsto.

## 6 IMPLEMENTACIÓN DE LA GESTIÓN DE OBSTÁCULOS

En esta sección se detalla la implementación de la funcionalidad destinada a gestionar los obstáculos que pueden generar sombras sobre las áreas designadas para la instalación de paneles solares. Para lograr esto, se desarrolló un sistema que permite a los usuarios indicar la presencia y altura de estos obstáculos dentro de la interfaz actual de diseño de instalaciones de módulos solares. Esta gestión de obstáculos representa un paso crucial en la mejora del diseño fotovoltaico, ya que permite a los usuarios considerar activamente el impacto de los elementos circundantes en el rendimiento de sus instalaciones solares.

La manera en la que se crean nuevos obstáculos en el diseño es muy similar a la ya existente para crear las cubiertas donde se posicionan los módulos fotovoltaicos. Esta consiste en que los usuarios seleccionan un área a partir de una serie de puntos en el mapa y posteriormente pueden añadir la altura correspondiente al obstáculo respecto al suelo mediante un *input*, como se puede observar en la figura 1. Los usuarios tienen la capacidad de agregar, mover y eliminar obstáculos directamente en el mapa interactivo proporcionado por la plataforma. Estos obstáculos pueden ser edificios, árboles u otros elementos que podrían proyectar sombras sobre los paneles solares. Además de crear esta funcionalidad para la gestión de obstáculos, se añade la posibilidad de ajustar la altura a las cubiertas de la instalación, como se puede ver en la figura 2, donde se posicionan las placas solares, de la misma forma que a los obstáculos, para posteriormente realizar los cálculos necesarios.

Para la creación de los obstáculos se utiliza la API de Google Maps para mostrar en el mapa la localización de la instalación fotovoltaica y se permite al usuario seleccionar una serie de puntos sobre este, creando así el polígono que define el obstáculo. Si nos fijamos en la figura 3 podemos ver los diferentes obstáculos en color amarillo y una cubierta con módulos fotovoltaicos en color azul.

La asignación de alturas a los diferentes elementos del diseño es una de las características principales que luego, en la sección 7, nos permite realizar el cálculo de sombras durante el año sobre los distintos módulos de la instalación. Para trabajar con estos datos, el sistema, internamente, utiliza la latitud y longitud de cada elemento para ajustar la posición y la altura en relación con las áreas seleccionadas para la instalación de paneles solares.

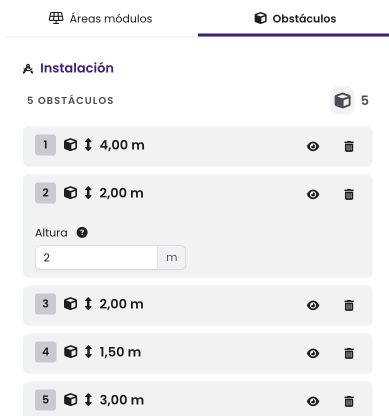


Fig. 1: Añadir altura a obstáculo



Fig. 2: Añadir altura a área

## 6.1. Modificaciones modelo de datos para gestión de obstáculos

Para soportar la funcionalidad de gestión de obstáculos en la plataforma, se realizaron modificaciones en el modelo de datos existente, específicamente en el objeto `layout` del modelo `PowerPlant`. Se añadió un nuevo atributo llamado `moduleKeepouts` para almacenar información detallada sobre los obstáculos, su ubicación y altura, y garantizar que esta información se pueda utilizar eficazmente en los cálculos de sombras y el diseño fotovoltaico. Además, se añadió otro atributo `surfaceHeight` en cada objeto `moduleArea` donde se almacena la altura establecida para el área. Se puede observar el diagrama Entidad-Relación con estas actualizaciones en la figura 14.

### 6.1.1. Atributo `moduleKeepouts`

El atributo `moduleKeepouts` se diseñó para almacenar una lista de objetos que representan los obstáculos. Cada objeto de esta lista contiene información detallada sobre los obstáculos, incluyendo `height`, que especifica la altura del obstáculo en metros; `polyshapeArray`, que es un array de objetos con las coordenadas geográficas (latitud y longitud) que definen el polígono del obstáculo, y `visible`, que es un booleano que indica si el obstáculo es visible en el diseño. Esta estructura permite un modelado preciso y detallado de los elementos que pueden afectar la instalación fotovoltaica.

## 7 ALGORITMOS Y FUNCIONALIDADES PARA EL CÁLCULO DE SOMBRAS

En esta sección se detallan los algoritmos y funcionalidades utilizados para calcular el porcentaje de sombras sobre



Fig. 3: Diferentes obstáculos en el diseño

cada módulo solar a lo largo del año. El objetivo principal es proporcionar a los usuarios datos precisos sobre el impacto de los obstáculos circundantes en la eficiencia de sus instalaciones fotovoltaicas.

Para lograr esto se implementaron varios algoritmos que simulan la trayectoria solar y evalúan la incidencia de sombras en cada momento del día y del año. Estos algoritmos tienen en cuenta la posición geográfica de la instalación, así como la altura y la geometría de los obstáculos identificados por los usuarios como se ha explicado en la sección 6.

El enfoque utilizado ha sido el trazado de rayos, que simula la propagación de la luz solar y calcula las áreas de sombra proyectadas por los obstáculos sobre los módulos solares. Este enfoque se basa en modelos de elevación digital (DEM) y datos geoespaciales para representar con precisión el terreno y los objetos circundantes [20].

Para simular la instalación en un mundo tridimensional se usa `Babylon.js`, partiendo de todos los datos de esta, concretamente la latitud y longitud, e incluyendo la inclinación, orientación y altura de cada área y de cada obstáculo. A partir de la simulación de la escena y los elementos en 3D, se calcula la posición del sol y, en consecuencia, la dirección de los rayos solares en cada momento del año. Tomando estos cálculos, se trazan varios rayos desde la superficie de cada módulo en la dirección del sol y se comprueba si estos chocan con algún elemento de la escena, como pueden ser otros módulos, cubiertas u obstáculos, en el trayecto. Por lo tanto, si el rayo trazado intersecciona con algún elemento significa que el punto desde donde se ha trazado el rayo está en sombra en ese momento concreto. De esta forma podemos calcular la sombra para cada módulo solar en cada momento del día de forma precisa.

### 7.1. Desarrollo de endpoints para el cálculo de sombras

Para facilitar la interacción con el sistema y permitir la ejecución de cálculos de sombras de manera eficiente, se desarrollaron varios endpoints en el backend. Estos endpoints permiten solicitar cálculos de sombras y recuperar los resultados de manera organizada y eficiente.

En particular, se creó un endpoint `POST` que permite a los usuarios iniciar el cálculo de sombras para una instalación específica descrito en la sección 7.2. Los resultados devueltos por este endpoint incluyen el porcentaje de sombras sobre cada módulo y otros datos relevantes, que son

esenciales para optimizar el diseño y la eficiencia de las instalaciones solares.

Además, se creó un endpoint *GET* para recuperar los datos del modelo *PowerPlantLayoutArea*, explicado en la sección 7.4. Este endpoint permite recuperar los datos de sombras, cosa que es crucial para analizar y ajustar los diseños según sea necesario, asegurando un rendimiento óptimo de las instalaciones fotovoltaicas.

## 7.2. Algoritmo de detección de obstáculos para el cálculo de sombras

A continuación, se presenta el algoritmo desarrollado para la detección de obstáculos y el cálculo de sombras 1.

---

### Algoritmo 1: Cálculo de Sombras

---

**Data:** Datos de la instalación: latitud, longitud.  
 Datos de cada área y obstáculo: inclinación, orientación y altura. Datos de cada módulo: posición e inclinación  
**Result:** Porcentaje de sombra sobre cada módulo solar, así como los resultados agrupados por áreas de módulos

```

begin
  for cada módulo solar  $M_i$  en la instalación do
    for cada hora  $H_j$  del año do
       $(\theta, \phi) \leftarrow \text{CalculateSunPosition}(H_j)$ ;
      for cada punto  $P_k$  en la superficie  $M_i$  do
        ray  $\leftarrow \text{TraceRay}(P_k, (\theta, \phi))$ ; for
          cada obstáculo  $O_l$  en la escena 3D do
            if  $\text{CheckIntersection}(\text{ray}, O_l)$  then
              Marcar  $P_k$  de  $M_i$  como en
              sombra en  $H_j$ ; Incrementar
              contador de sombras para
               $M_i$  en  $H_j$ ;
            Calcular porcentaje de sombra anual para
             $M_i$ ;
          SaveShadowData();

```

---

Es importante destacar que no se comprueban todos los puntos de la superficie de cada módulo para la intersección con obstáculos. En su lugar, se seleccionan puntos específicos a lo largo de las diagonales del módulo y en el centro. Este enfoque reduce el número de cálculos necesarios, mejorando el rendimiento del algoritmo, sin sacrificar significativamente la precisión del cálculo de sombras. La selección de puntos está diseñada para capturar la variabilidad de la sombra en la superficie del módulo de manera eficiente.

Posteriormente, en la sección 7.3, se presentarán pruebas de rendimiento para evaluar la eficacia de este método de selección de puntos en comparación con un enfoque que consideraría mayor o menor número de puntos de la superficie del módulo.

## 7.3. Pruebas de rendimiento

Para evaluar la eficiencia del algoritmo de cálculo de sombras, se realizaron pruebas variando el número de subdivisiones utilizadas para determinar los puntos de intersección en la superficie de cada módulo solar. A continuación, en la figura 4, se presentan los resultados obtenidos mediante un gráfico que ilustra el tiempo de ejecución en función del número de subdivisiones.

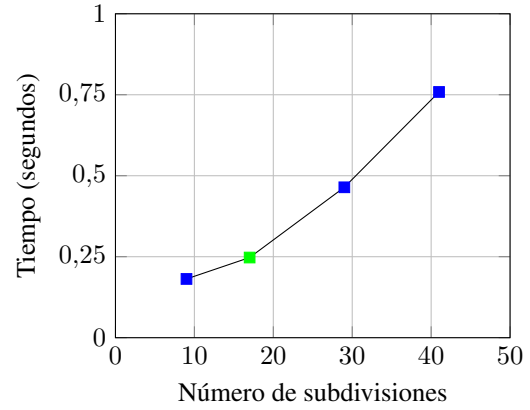


Fig. 4: Rendimiento cálculo de sombras

El algoritmo implementado selecciona puntos específicos en la superficie de cada módulo solar, estos se encuentran a lo largo de las diagonales y en el centro del módulo, en lugar de verificar todos los puntos. Esta estrategia optimiza el rendimiento sin comprometer significativamente la precisión del cálculo de sombras.

Adicionalmente, para evitar cálculos innecesarios, se tiene en consideración la salida y puesta del sol, lo que permite limitar los tiempos del día en los que se realizan los cálculos de sombra. Si el sol está por debajo del horizonte, no se realizan cálculos de sombra para esos periodos, optimizando aún más el rendimiento del algoritmo.

Estos resultados y optimizaciones demuestran la eficiencia del enfoque adoptado, proporcionando un balance adecuado entre precisión y tiempo de ejecución, lo cual es crucial para aplicaciones prácticas en instalaciones solares.

Cabe mencionar que estas pruebas fueron realizadas sobre una instalación residencial, concretamente la misma que se muestra en todas las figuras de este documento. Esta consta de una única área, 17 módulos solares y 5 obstáculos.

En base a las pruebas de rendimiento, se decidió comprobar 17 puntos por cada módulo. Este número fue seleccionado después de analizar el equilibrio entre la precisión y el tiempo de ejecución, puesto que se observó que, al superar este *threshold*, el tiempo de cálculo aumentaba considerablemente sin que la precisión en la determinación de las sombras mejorara significativamente. Esta elección asegura que el sistema sea eficiente y preciso, adecuado para cualquier tipo de instalación, donde se busca maximizar la eficiencia sin incrementar demasiado el tiempo de cálculo.

## 7.4. Almacenamiento de datos en nuevo modelo

Los resultados del cálculo de sombras se almacenan en un nuevo modelo de datos estructurado que permite su consulta y análisis posterior llamado *PowerPlantLayoutArea*.



El modelo utilizado está implementado en LoopBack versión 3.x usando MongoDB como base de datos, y se define como se muestra en el modelo de la figura 13.

El campo `shadowsAnnualAverage` mide el porcentaje promedio anual de sombras que afectan a la instalación, ayudando a evaluar su eficiencia energética. Por otra parte, `shadowsMonthlyData` proporciona datos mensuales y horarios de sombras, detallando la media mensual (`shadowsMonthlyAverage`) y los resultados horarios (`shadowsHourlyData`). En último lugar, `shadowsModuleData` incluye el porcentaje de sombras anual para cada módulo mediante el atributo `shadowsModuleAnnualAverage`.

Finalmente hay que señalar que en el modelo *PowerPlant* se ha creado una nueva propiedad `totalShadows` para almacenar el valor anual de sombras total resultante del agregado de las diferentes áreas de módulos solares pertenecientes a una instalación concreta. El almacenamiento de estos datos facilita su integración en la plataforma, permitiendo a los usuarios acceder a informes detallados y visualizaciones interactivas de las sombras.

7.5. Visualización de datos en la plataforma

La visualización de los datos calculados en la plataforma se ha desarrollado utilizando Angular y AngularJS. Se ha implementado una arquitectura que incluye varios componentes y servicios para gestionar y mostrar los datos de manera eficiente.

7.5.1. Arquitectura

- **Servicio de datos:** Se ha desarrollado un servicio en Angular para guardar todos los datos recuperados del backend. Este servicio se encarga de gestionar la comunicación con el servidor y almacenar los datos necesarios para la visualización.
- **Componente de tabla:** Se ha creado un componente en Angular para mostrar los datos en forma de tabla. Este componente utiliza los datos proporcionados por el servicio de datos y los presenta en una interfaz gráfica intuitiva, facilitando la interpretación y análisis de los mismos.
- **Visualización en el mapa interactivo:** Para pintar los módulos en el mapa, se ha creado un servicio utilizando la API de Google Maps. Este servicio configura las opciones de los polígonos con los colores correspondientes a los porcentajes de sombra calculados. Se ha diseñado una paleta de colores con diferentes niveles (steps) para representar visualmente el porcentaje de sombra en cada módulo.

7.5.2. Interfaz gráfica

La interfaz gráfica de la plataforma incluye las siguientes características:

- **Mapas de calor:** Se utiliza para indicar las zonas de sombra en los módulos solares, siguiendo una leyenda de colores, proporcionando una representación visual clara y comprensible de la distribución de sombras. Asimismo, al hacer *hover* encima de un módulo en el

mapa, aparece un *tooltip* con el porcentaje de sombras anual del módulo concreto. Como se puede observar en la figura 5.

- **Tabla de porcentaje de sombra:** Esta tabla muestra el porcentaje de sombra a lo largo de cada hora de cada mes durante el año, permitiendo un análisis detallado del comportamiento de las sombras en diferentes periodos. También figura el porcentaje de sombras en cada mes y el agregado total de sombras para cada área. Como se puede observar en la figura 6.



Fig. 5: Afectación sombras en los módulos

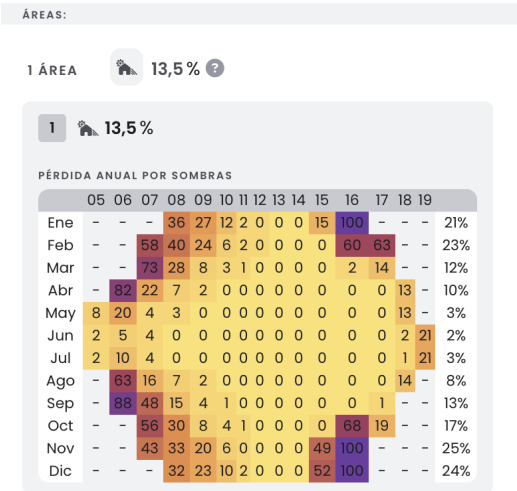


Fig. 6: Tabla horaria porcentaje afectación de sombras

El desarrollo de esta arquitectura en Angular y AngularJS, junto con la integración de la API de Google Maps y la implementación de una paleta de colores específica, ha permitido crear una plataforma robusta y eficiente para la visualización y análisis de los datos de sombra en instalaciones solares.

7.6. Comunicación en tiempo real usando WebSockets

Para lograr una actualización en tiempo real de los datos de sombras y permitir la interacción dinámica con la plataforma, se ha implementado un sistema de comunicación basado en WebSockets, utilizando Deepstream. Esto permite

a los usuarios ver cambios inmediatos en las sombras cuando se ajustan parámetros de diseño como la posición o la orientación de los módulos solares.

La inicialización de la conexión WebSocket se realiza a través del método que establece la conexión con el servidor y configura los gestores de eventos necesarios para manejar la comunicación.

Una vez establecida la conexión WebSocket, desde el cliente es necesario suscribirse a los eventos para gestionar el progreso del cálculo. Al suscribirnos nos encargamos de enviar un mensaje de suscripción al servidor y de manejar los mensajes recibidos, permitiendo así la actualización en tiempo real de la interfaz de usuario. Cada notificación es recibida y gestionada para mostrar al usuario, ya sea actualizando una barra de progreso o mostrando los datos recibidos. El algoritmo 2 detalla el proceso de suscripción y manejo de mensajes.

#### Algoritmo 2: Suscripción a eventos WebSocket

**Data:** WebSocket (*webSocket*), Tipo de evento (*eventType*)

**Result:** Suscripción a eventos y manejo de mensajes recibidos

**begin**

```
subscriptionMessage ← { "type":
  eventType, "action": "subscribe" }
sendMessage (webSocket,
  subscriptionMessage)
while recibir mensaje de webSocket do
  message ← esperar mensaje
  handleMessage (message)
```

El flujo de trabajo con WebSockets se detalla en el diagrama de secuencia de la figura 7.

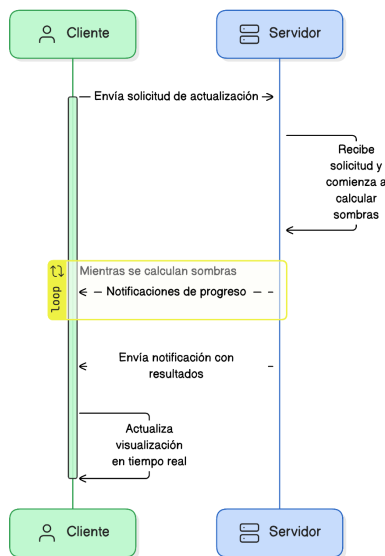


Fig. 7: Diagrama de secuencia Websockets

Este sistema de comunicación en tiempo real asegura que los usuarios puedan ver inmediatamente los efectos de sus modificaciones en el diseño de la instalación solar, mejorando la experiencia de usuario y la eficiencia en el proceso de diseño.

## 7.7. Auto-cálculo implementado al modificar el diseño

Una de las funcionalidades más avanzadas de este proyecto es el auto-cálculo de sombras cuando se modifican los parámetros de diseño en la plataforma. Cada vez que un usuario realiza cambios en el diseño, como mover un módulo solar o ajustar su inclinación, el sistema automáticamente recalcula las sombras y actualiza los resultados en la interfaz gráfica.

El diagrama de flujo de la figura 8 ilustra el proceso completo del auto-cálculo de sombras al modificar el diseño.

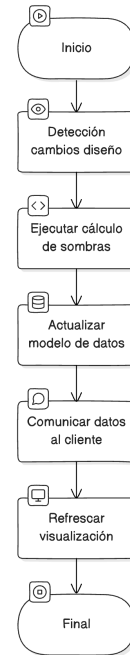


Fig. 8: Diagrama de flujo auto-cálculo de sombras

## 8 DESARROLLO DEL VISOR 3D

El desarrollo del visor 3D para la plataforma de diseñador fotovoltaico ha sido una parte fundamental del proyecto. Utilizando la biblioteca Babylon.js, se ha creado una interfaz intuitiva y altamente interactiva que permite a los usuarios visualizar su instalación fotovoltaica en un entorno tridimensional. En esta sección, se detallan las características y funcionalidades implementadas en el visor 3D.

La vista principal del visor 3D muestra la disposición tridimensional de la instalación fotovoltaica. Los usuarios pueden explorar la escena, rotarla y hacer zoom para obtener una vista completa de su diseño. Cada módulo solar y obstáculo se representa con precisión en su ubicación correspondiente en el terreno, lo que permite una visualización detallada de la disposición y configuración de los componentes, como se puede observar en la figura 9.

Una funcionalidad destacada del visor 3D es la capacidad de visualizar cómo afectan las sombras a la instalación fotovoltaica. En esta vista, los módulos solares se pintan con diferentes colores según su nivel de sombreado, lo que proporciona a los usuarios una representación visual clara del impacto de los obstáculos en la eficiencia de su sistema fotovoltaico (figura 15). Esta representación es consistente

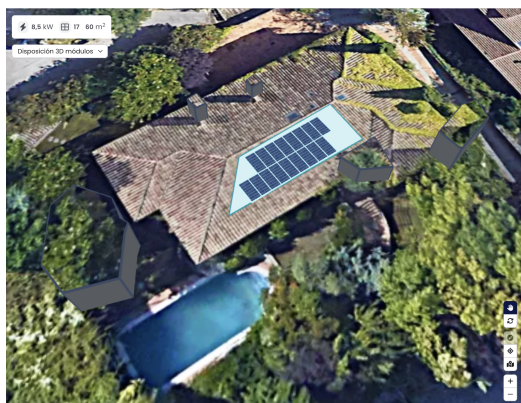


Fig. 9: Disposición instalación en visor 3D

con la vista proporcionada en el diseñador 2D, lo que facilita la comparación entre ambas interfaces.

Otra funcionalidad clave del visor 3D es la capacidad de proyectar sombras en tiempo real sobre los módulos solares para cualquier hora, día y año específicos. Los usuarios pueden seleccionar una fecha y hora determinadas y observar cómo se proyectan las sombras sobre los módulos solares en ese momento preciso (figura 16). Esto proporciona una comprensión más profunda del comportamiento de la instalación a lo largo del día y del año, lo que permite realizar ajustes precisos en el diseño para optimizar su rendimiento.

### 8.1. Menú de acciones

En todas las vistas del visor 3D, hay disponible un menú con diferentes acciones como se puede ver en la figura 10, enumeradas de arriba a abajo, que facilitan la interacción y el control sobre la visualización: cambiar modo de navegación por el mapa (mover o rotar), orientar hacia el norte, centrar en el área de módulos, centrar en la ubicación seleccionada al crear la instalación, y hacer zoom in y zoom out.

Este menú proporciona a los usuarios un acceso rápido a las funcionalidades de navegación y visualización, mejorando así la experiencia de uso del visor 3D. En conjunto, estas características ofrecen a los usuarios una herramienta poderosa para visualizar, analizar y optimizar sus diseños fotovoltaicos, facilitando la toma de decisiones informadas durante la fase de diseño.

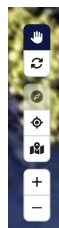


Fig. 10: Menú de acciones visor 3D

## 9 INTEGRACIÓN DE SOLAR API

La integración de Solar API en el visor 3D del diseñador fotovoltaico introduce una funcionalidad avanzada que permite a los usuarios analizar detalladamente la exposición

solar de su instalación. Solar API, proporcionado por Google, ofrece datos precisos sobre la radiación solar en diversas ubicaciones geográficas, facilitando así una evaluación exhaustiva del potencial de generación de energía solar.

Esta integración se implementa mediante una nueva vista en el visor 3D, que muestra la exposición solar del terreno utilizando los datos de Solar API. Se han incorporado dos modos de visualización: uno que muestra la exposición solar de todo el terreno y otro que se centra exclusivamente en los edificios presentes en la instalación. Esta funcionalidad permite a los usuarios seleccionar áreas específicas para un análisis más detallado, optimizando así el diseño de la instalación. Además de la visualización de la exposición solar, Solar API proporciona información sobre la radiación máxima recibida por la instalación. Este dato es fundamental para evaluar el potencial de generación de energía solar en el área seleccionada, y permite a los diseñadores tomar decisiones informadas sobre la ubicación y orientación de los paneles solares.

En la figura 11 se presenta una captura de pantalla que muestra la exposición solar en el visor 3D, mientras que la figura 17 ilustra la exposición solar centrada exclusivamente en los edificios.

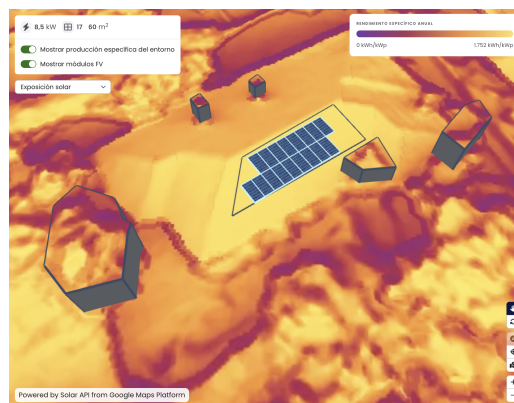


Fig. 11: Exposición solar en visor 3D con datos Solar API

## 10 CONCLUSIONES

Este proyecto ha logrado importantes avances en la optimización del diseño de instalaciones fotovoltaicas en la plataforma Wattwin, cumpliendo con los objetivos planteados. Se ha mejorado significativamente la precisión del diseño mediante la implementación de una funcionalidad que permite definir obstáculos y calcular las sombras proyectadas sobre los paneles solares a lo largo del año. Esta mejora aborda directamente el objetivo de considerar elementos que generan sombras en el diseño de instalaciones fotovoltaicas, permitiendo a los usuarios indicar la presencia y altura de obstáculos como edificios, árboles u otros elementos colindantes.

Además, se ha desarrollado un visor 3D que permite visualizar en tiempo real el impacto de las sombras y la disposición de los módulos, facilitando una mejor toma de decisiones. Esto cumple con el objetivo de crear un visor 3D para la simulación de instalaciones fotovoltaicas, proporcionando una herramienta visual que mejora la comprensión y el análisis del diseño.



La integración de la Solar API de Google ha enriquecido la plataforma con datos precisos sobre la radiación solar, cumpliendo con el objetivo de incorporar datos y funcionalidades de esta API. Esta integración aporta información valiosa que optimiza el diseño al considerar factores relevantes para la eficiencia energética.

Para el futuro, se propone optimizar aún más los algoritmos de cálculo de sombras y ampliar las fuentes de datos geoespaciales, lo cual mejorará la precisión del modelado del entorno y la proyección de sombras. También se prevé agregar funcionalidades adicionales al visor 3D, como la simulación de condiciones climáticas variables y el análisis financiero basado en la producción energética esperada.

En conclusión, este proyecto ha cumplido con éxito los objetivos establecidos, proporcionando herramientas más precisas y funcionales para el diseño de instalaciones fotovoltaicas. Las mejoras realizadas y las futuras líneas de desarrollo posicionan a la plataforma Wattwin como una herramienta líder en el diseño de sistemas solares, contribuyendo significativamente a la transición hacia energías renovables y sostenibles.

## AGRADECIMIENTOS

Me gustaría expresar mi más profundo agradecimiento a todas las personas que han contribuido a la realización de este proyecto.

En primer lugar, quiero dar las gracias a mi tutor en la universidad, Adrià Figuerola Torrell, por su constante orientación y asesoramiento académico.

Asimismo, extendiendo mi agradecimiento a mi tutor en la empresa, Holger Castillo Cajape, por su guía y apoyo a lo largo de todo el desarrollo de este trabajo. Su experiencia y conocimientos han sido fundamentales para superar los desafíos y alcanzar los objetivos propuestos.

## REFERENCIAS

- [1] (2023) Renewables 2023. IEA. [Online]. Available: <https://www.iea.org/reports/renewables-2023>
- [2] UNEF, “10 datos que demuestran que la energía solar es una oportunidad para España,” *UNEF*, 2024, 14-Feb-2024.
- [3] (2022) Wattwin - integra todas las herramientas que necesitas para gestionar y escalar tu negocio de autoconsumo solar. Wattwin. [Online]. Available: <https://wattwin.com/>
- [4] Solar api overview. Google for Developers. [Online]. Available: <https://developers.google.com/maps/documentation/solar/overview>
- [5] ¿qué es scrum? [+ cómo empezar]. Atlassian. [Online]. Available: <https://www.atlassian.com/es/agile/scrum>
- [6] P. Letelier Torres, *Una actividad para enseñar el uso de tableros kanban y diagramas de flujo acumulado*. Universitat Oberta La Salle, Jul 2015, p. 288–295. [Online]. Available: <http://hdl.handle.net/2117/77466>
- [7] A. R. León Yacelga and M. A. Checa Cabrera, “Uso de tableros kanban como apoyo para el desarrollo de las metodologías ágiles,” *Universidad y Sociedad*, vol. 14, no. S2, pp. 208–214, abr. 2022. [Online]. Available: <https://rus.ucf.edu/cu/index.php/rus/article/view/2760>
- [8] Software de colaboración para equipos de software, ti y empresa. Atlassian.com. [Online]. Available: <https://www.atlassian.com/es>
- [9] J. Loeliger and M. McCullough, *Version Control with Git: Powerful tools and techniques for collaborative software development*. .o'Reilly Media, Inc., 2012.
- [10] Atlassian, “Sourcetree,” 2024. [Online]. Available: <https://www.sourcetreeapp.com/>
- [11] —, “Jira software,” 2024. [Online]. Available: <https://www.atlassian.com/es/software/jira>
- [12] —, “Confluence software,” 2024. [Online]. Available: <https://www.atlassian.com/es/software/confluence>
- [13] Pvsyst. PVsyst SA. [Online]. Available: <https://www.pvsyst.com/>
- [14] Helioscope. Folsom Labs. [Online]. Available: <https://app.helioscope.com/>
- [15] Three.js – javascript 3d library. Threejs.org. [Online]. Available: <https://threejs.org/>
- [16] Babylon.js: Powerful, beautiful, simple, open - web-based 3d at its best. Babylon.js. [Online]. Available: <https://www.babylonjs.com/>
- [17] T. Whitted, “An improved illumination model for shaded display,” in *ACM SIGGRAPH 2005 Courses*, ser. SIGGRAPH '05. New York, NY, USA: Association for Computing Machinery, 2005, p. 4–es. [Online]. Available: <https://doi.org/10.1145/1198555.1198743>
- [18] M. Pharr and G. Humphreys, *Physically Based Rendering: From Theory to Implementation*. Morgan Kaufmann, 2016.
- [19] J. P. Wilson and J. C. Gallant, *Terrain Analysis: Principles and Applications*. John Wiley & Sons, 2000.
- [20] GeoSolutions Consulting, “Modelos de elevación digital,” 2024, accessed: 2024-06-13. [Online]. Available: <https://www.geosolutionsconsulting.com/modelos-de-elevacion-digital/>

APÉNDICE

A DIAGRAMA DE GANTT



Fig. 12: Diagrama de Gantt

B DIAGRAMA ENTIDAD-RELACIÓN MODELO SOMBRAS

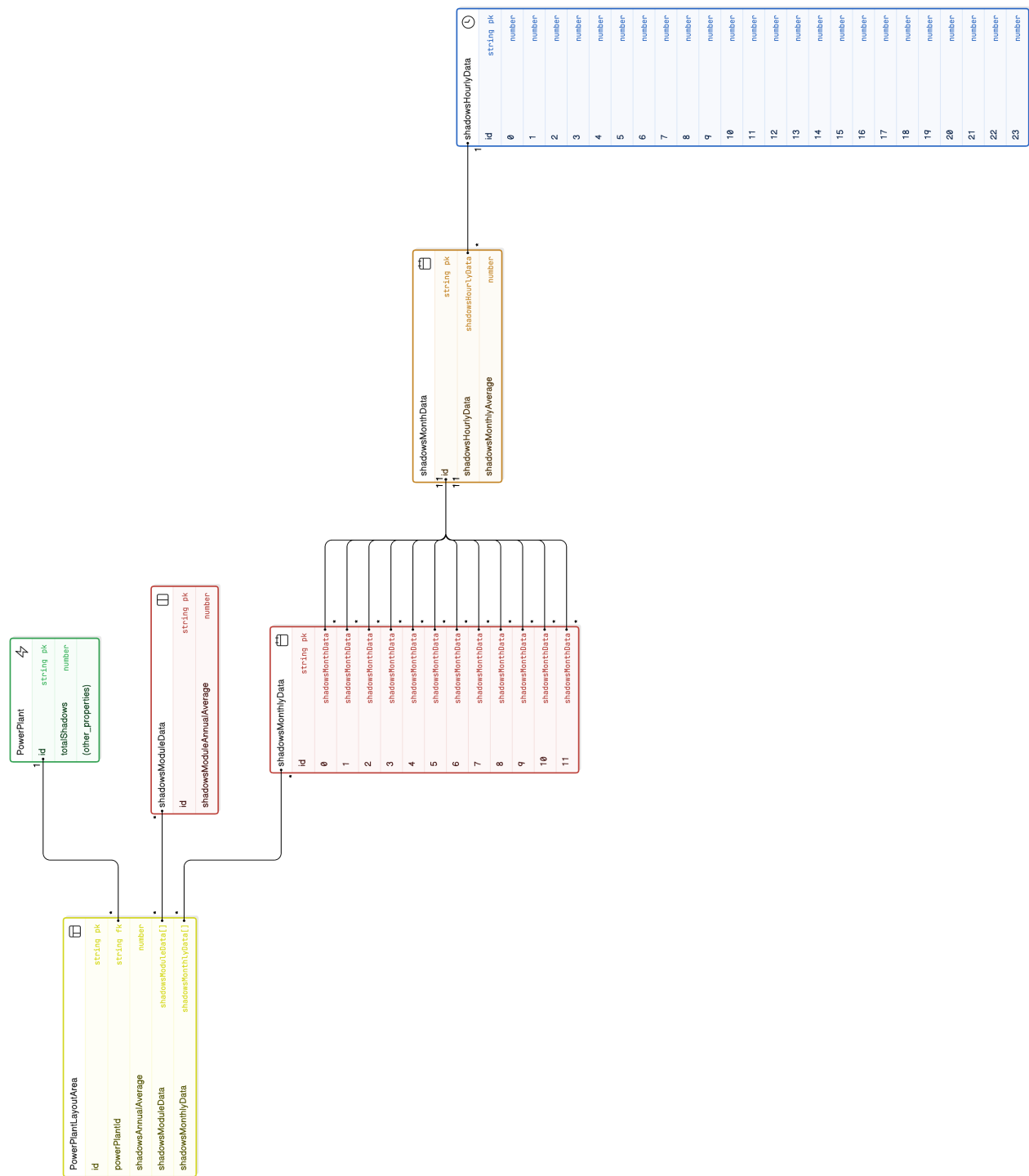


Fig. 13: Diagrama Entidad-Relación modelo datos sombras

C DIAGRAMA ENTIDAD-RELACIÓN MODELO OBSTÁCULOS

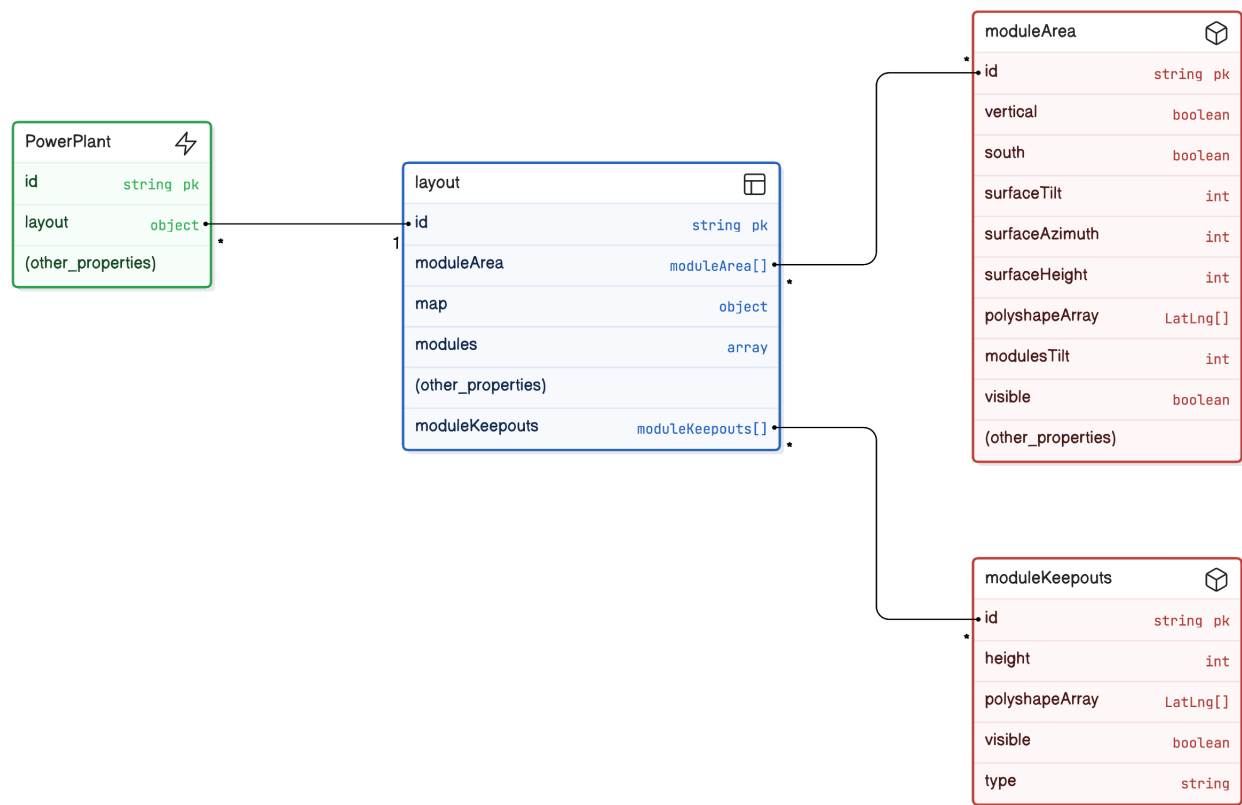


Fig. 14: Diagrama Entidad-Relación modelo actualizado para almacenar obstáculos y alturas

D IMÁGENES VISOR 3D

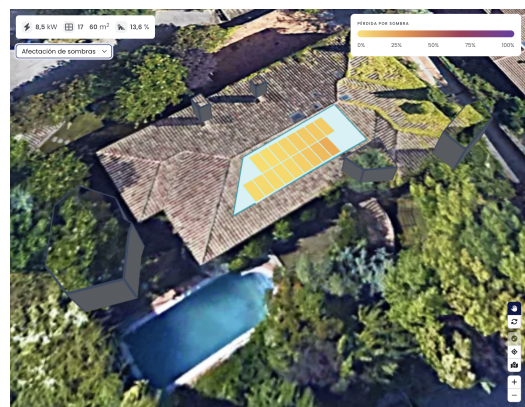


Fig. 15: Afectación de sombras en visor 3D

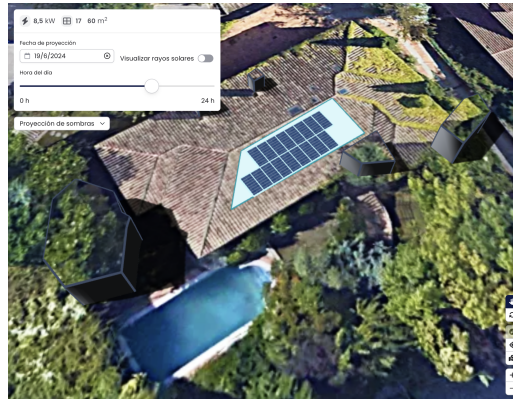


Fig. 16: Proyección de sombras en visor 3D



Fig. 17: Exposición solar únicamente edificios en visor 3D con datos Solar API