

Apphelion, una aplicación para simular la trayectoria solar y la incidencia lumínica sobre viviendas

Dr. Matías Micheletto

Sendevo Software, Bahía Blanca (8000), Buenos Aires, Argentina.

holasendevo@gmail.com

Resumen

En términos de arquitectura, la orientación de viviendas y edificaciones es fundamental para mejorar aspectos relacionados con la eficiencia energética, la comodidad habitacional y el aprovechamiento de la luz natural. Otros factores de diseño que pueden influir en estos aspectos pueden ser la ubicación de plantaciones o la extensión de techos y tejados mediante aleros, también denominados tejaroces o socarrenes. Para comprender la trayectoria solar y la incidencia de la misma sobre, no sólo las construcciones sino también paneles solares, en particular en latitudes alejadas del ecuador terrestre y durante las distintas épocas del año, el cambio del ángulo de incidencia del sol puede variar de manera considerable. En este artículo se detallan las fórmulas empleadas en cartas solares para calcular determinísticamente la posición del sol en el cielo en función de la ubicación geográfica y la fecha y hora. Se presenta la implementación de un software que permite simular la trayectoria solar y visualizar su efecto sobre modelos tridimensionales.

1. Introducción

La trayectoria solar se refiere al camino que sigue el sol a lo largo del día y a lo largo del año. Este recorrido varía según la ubicación geográfica y la estación del año. Comprender la trayectoria solar permite a los arquitectos y diseñadores optimizar la orientación de los edificios para maximizar el aprovechamiento de la luz natural y minimizar la dependencia de la iluminación artificial y los sistemas de climatización.

La correcta orientación de una construcción puede aumentar significativamente la eficiencia energética [6, 7], reduciendo los costos de calefacción, refrigeración e iluminación. Por ejemplo, orientar las ventanas principales hacia el sur en el hemisferio norte y hacia el norte en el hemisferio sur permite capturar la mayor cantidad de luz solar durante el invierno, mientras que al mismo tiempo se evita el sobrecalentamiento en verano, siempre y cuando se disponga de extensiones en el techo a modo de aleros, para crear sombra sobre estas aberturas.

En el caso de paneles solares ocurre algo similar. Para aprovechar al máximo la generación de energía eléctrica, los vectores normales de sus superficies de captación debe apuntar hacia la posición del sol [8]. También es posible emplear dispositivos de seguimiento solar, los cuales cuentan con sensores que detectan la posición del sol, para orientar los paneles automáticamente. En el caso de disponer los datos de trayectoria solar en función de la época del año, puede evitarse el empleo de sensores, incluso de sistemas automáticos, sino que se ajusta manualmente la inclinación del panel en función de la estación, el mes o incluso semana [5].

En este artículo se resumen los cálculos que intervienen en la determinación de la posición del sol en función de la ubicación e instante del año en que se encuentra cierto observador y la implementación de estas ecuaciones en un software multiplataforma para simular la incidencia del sol sobre superficies de modelos tridimensionales.

2. Carta solar

La posición del sol en el cielo tal como la ve un observador en la tierra depende de varios factores como la ubicación geográfica, la fecha y la hora del día. La grilla terrestre, demarcada mediante trazados verticales y horizontales a lo largo del globo, permite referenciar ubicaciones en cualquier parte del planeta. La medida de los ángulos longitudinales respecto a la hora del día tiene que ver con ciertas convenciones que fueron definidas históricamente, por lo que para realizar los cálculos pertinentes es necesario incorporar una carta tabulada a modo de contar con todos los datos.

En astronomía, el ángulo de declinación del sol es la distancia angular norte o sur del ecuador celeste. Una de las fórmulas utilizadas para calcular la declinación del sol está dada por

$$\delta = 23,45 \cdot \sin \left(\frac{360}{365,25} \cdot (N + 10) \right), \quad (1)$$

donde

- δ es el valor de declinación en grados.
- N es el número de días desde el comienzo del año.

Por otro lado, el acimut del sol es la distancia angular a lo largo del horizonte medido en sentido horario desde el norte. Este ángulo va cambiando a lo largo del día a medida que el sol cruza el cielo. El acimut se puede calcular, para el caso del sol, a partir de la siguiente fórmula,

$$\cos(\theta) = \frac{\sin(\delta) \cdot \cos(\phi) - \cos(H) \cdot \cos(\delta) \cdot \sin(\phi)}{\sin(\theta_s)}, \quad (2)$$

donde

- θ es el ángulo cenital solar ($90^\circ - \text{ángulo de elevación solar}$).
- δ es la declinación del sol.
- ϕ es la latitud del observador.
- H es el ángulo horario del sol que se calcula como la diferencia de tiempo por 15° ($\Delta T \cdot 15^\circ$).
- θ_s es el coseno del ángulo cenital solar.

Existen modelos más complejos que incluyen otros factores para incorporar correcciones de manera de obtener estimaciones más exactas. A los efectos de la funcionalidad que se pretende cubrir con la aplicación en cuestión, las aproximaciones presentadas aquí son suficientes.

3. Materiales y métodos

Pensando en lograr compatibilidad multiplataforma, la aplicación se construye íntegramente mediante lenguaje de programación web. Por lo que la elección de la tecnología se basa no sólo en la portabilidad sino en que permite incorporar tanto la implementación de la simulación como la capa de visualización de datos en una misma base de código.

El proyecto se organiza en distintas etapas. Se parte de un núcleo para contener el modelo de negocio, el cual incluye la implementación de los algoritmos de simulación. En un esquema de *Clean Architecture* [2], este esquema se incluye en la capa central, donde son definidas las entidades del modelo.

A continuación se define una capa para el diseño de la interfaz gráfica, donde se separa la visualización de la información en tres secciones, una para la selección de la ubicación, otra para visualizar la carta solar y una tercera para simular la iluminación solar sobre un modelo 3D. En la parte superior se cuenta con un menú compuesto por todos los controles necesarios para la simulación, así como los desplegables para establecer ciertos parámetros relacionados con la configuración de la visualización o la selección de la fecha y hora.

Finalmente se programó un módulo para la importación de un modelo 3D y la renderización del mismo en la escena de simulación. En las siguientes subsecciones se detallan las principales consideraciones al momento de realizar la implementación de este software.

3.1. Algoritmos

El código de implementación de los algoritmos de cálculo se dividen en distintas secciones dependiendo de la funcionalidad. El dato de entrada que requiere el algoritmo de cálculo incluye principalmente la ubicación geográfica. A partir de esta información se calcula la trayectoria solar a lo largo de todo el año mediante las ecuaciones 1 y 2, haciendo un barrido de muestreo por las distintas fechas y horarios. Esto permite generar la tabla solar anual y la tabla solar diaria, cuyos datos son empleados para generar los gráficos de horas de luz, las trayectorias y el cálculo de proyecciones de sombras, entre otros.

Realizar el procesamiento de la lista de datos es computacionalmente intensivo y al ejecutar esta rutina como proceso de la misma aplicación, provoca demora y bloqueo de la aplicación. Una manera de evitar eso es usar *service workers*, los cuales son procesos que se ejecutan en segundo plano. Una manera de acelerar la ejecución de los procesos en paralelo, es empleando rutinas de *WebAssembly* (*WASM*, por sus siglas en inglés), el cual es un lenguaje de bajo nivel que el navegador puede ejecutar a mayor velocidad que con *JavaScript*.

Si bien se puede generar código *WASM* a partir de la compilación de varios lenguaje de programación como *Rust*, *C/C++* o *AssemblyScript*, por una cuestión de familiaridad con el entorno, se eligió *C/C++* y *Emscripten*¹ como compilador.

Los módulos compilados se pueden importar directamente desde el código web para intercambiar información en varios formatos, lo que permite enviar los datos de ubicación geográfica y recibir los arreglos con datos de la trayectoria solar. Este esquema acelera el procesamiento y mejora la experiencia del usuario, al calcular la carta solar a medida que se desplaza la ubicación sobre el mapa.

3.2. Visualización

La interfaz gráfica se implementó con distintas librerías. Con *jQuery*² se realiza el intercambio de variables ente los métodos de procesamiento y los elementos del documento, como los controles y formularios. Para la definición de estilos se utilizó *Bootstrap*³, que provee un esquema completo para estilizar y alinear componentes de la ventana, entre otras funciones.

El mapa que muestra la ubicación sobre la que se calcula la carta solar se realizó con *Leaflet*⁴, que provee un marco para mostrar información geográfica extraída de *OpenStreetMap*⁵.

Para ilustrar el modelo 3D de construcciones como casas u otras edificaciones, se valió de la librería *Three.js*⁶ y finalmente, para la presentación de la gráfica que muestra la duración del día se utilizó *D3*⁷.

3.3. Importación de modelos 3D

Una función relevante de la aplicación es la posibilidad de importar modelos 3D de construcciones. Los formatos aceptados son OBJ, PLY o STL, de los cuales, los primeros dos son preferibles al contar con información de texturas y colores. Los archivos importados son procesados para ser convertidos al formato *JSON* que utiliza la librería *Three.js*.

3.4. Distribución

La distribución del software se refiere a los medios por los cuales el mismo puede ser accedido por los distintos usuarios. Por cuestiones de seguridad, siempre se utilizan sitios de descarga seguros, los cuales implican direcciones de *URL* de reputación comprobada, como tiendas de aplicaciones oficiales o sitios web con certificado *SSL*.

El acceso al software *Apphèlion* se puede realizar a través de tres medios disponibles: el sitio web de la aplicación con posibilidad de instalación como *PWA*, la tienda de aplicaciones de *Android*® y por medio de archivos ejecutables tanto para *Windows*® como para *Linux* accesible a través de la

¹<https://emscripten.org/>

²<https://jquery.com/>

³<https://getbootstrap.com/>

⁴<https://leafletjs.com/>

⁵<https://www.openstreetmap.org/>

⁶<https://threejs.org/>

⁷<https://d3js.org/>

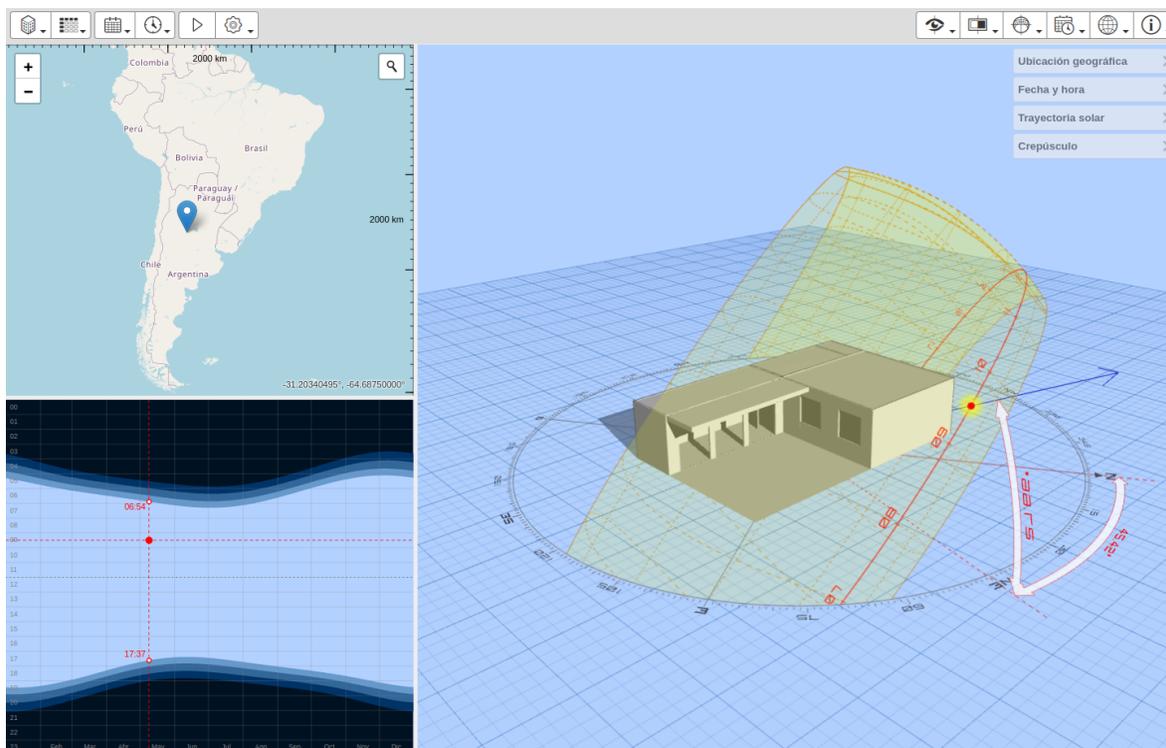


Figura 1: Captura de pantalla.

página oficial del desarrollador. Para el caso de la compilación de la aplicación al formato ejecutable de *Android*® (.apk), se utiliza un esquema híbrido, el cual puede ser llevado a cabo con el *framework Capacitor.js*⁸. Para el caso de aplicaciones simples, este es un esquema que ha resultado práctico en anteriores desarrollos [1, 4, 3]. La aplicación puede ser luego distribuida en la tienda de aplicaciones *Google Play*®. De manera similar, se emplea *Electron.js*⁹ para la compilación del software para PC.

3.5. Branding

La etimología del producto se basa en la combinación de las palabras “App” que es la abreviatura de aplicación y “Helion”, que es una variación de “Helios”, quien fuera un personaje de la mitología griega que simboliza el sol. En otros casos como en el contexto de la astronomía, la palabra “Helion” hace referencia al cuerpo celeste alrededor del cual orbita la Tierra. Respecto del diseño de logo e identidad del software Apphelion, se emplea, a modo de ícono, una imagen simplificada del sol en un amanecer entre edificios de ciudad. A la fecha de redacción de este trabajo, la interfaz gráfica presenta un estilo de tema claro, sin la opción de alternar a un tema oscuro.

En la figura 2 se muestra el logotipo diseñado para la aplicación, que se encuentra presente como ícono de página (favicono) de la versión web, una captura de pantalla de la versión móvil y una captura de pantalla del encabezado de la página de aterrizaje o *landing page*, la cual ilustra las principales características de la aplicación y el flujo de trabajo.

4. Conclusión

La implementación de los cálculos de trayectoria solar en un software como el planteado en este artículo y sumado a la posibilidad de renderizar un modelo 3D para visualizar la iluminación natural sobre el mismo, hacen de esta herramienta un sistema sustancialmente útil durante el diseño de una vivienda, como su orientación, la dirección y dimensión de aberturas, las estructuras de sombra como aleros, cobertizos y árboles, entre muchas otras.

⁸<https://capacitorjs.com/>

⁹<https://www.electronjs.org/es/>

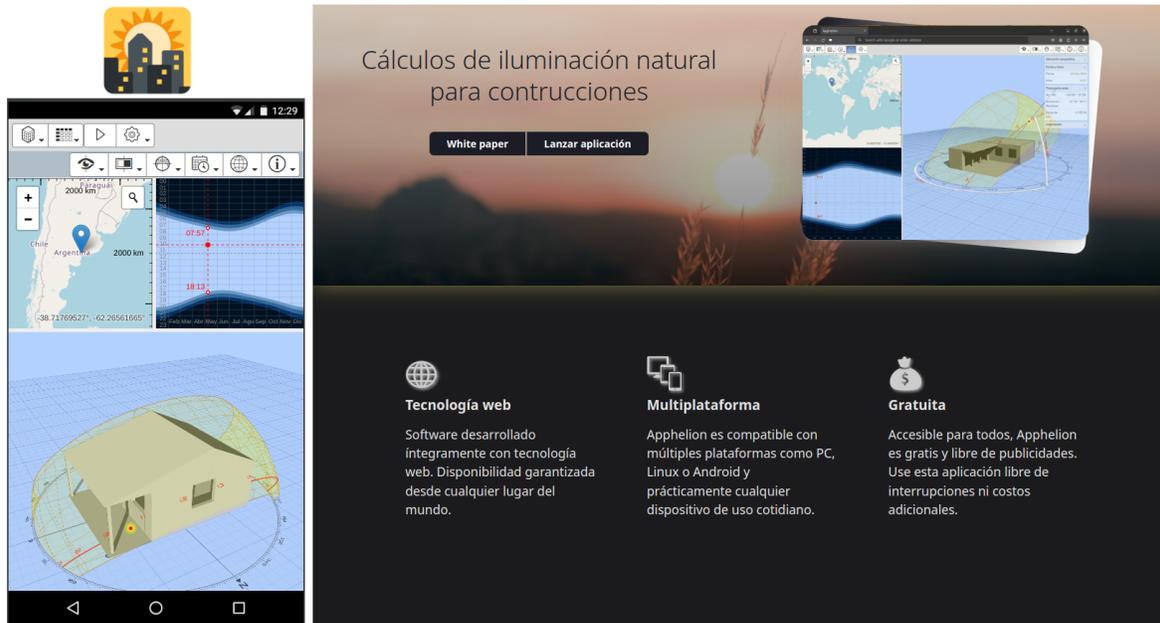


Figura 2: Logotipo (Arriba a la izquierda), app móvil (abajo a la izquierda) y *landing page* (derecha).

La elección de la tecnología web como base para el proyecto Apphelson se basa en que ofrece una serie de beneficios significativos para la implementación de software. Desde sus inicios, se enfoca en la accesibilidad y compatibilidad, permitiendo a los usuarios acceder al software desde cualquier dispositivo con conexión a Internet. Esto fomenta la colaboración y la flexibilidad en el trabajo. Además, la tecnología web simplifica el proceso de actualización y mantenimiento, ya que los cambios pueden implementarse de forma centralizada y sin necesidad de descargar actualizaciones manualmente. Asimismo, reduce los costos de infraestructura, ya que no se requieren instalaciones complejas en los dispositivos de los usuarios.

A modo de trabajo futuro resta implementar los cálculos de radiación para determinar la cantidad de energía solar disponible sobre cada una de las superficies del modelo 3D. Si bien es un factor que depende en buena medida de los materiales de dichas superficies, es una herramienta más que aporta certezas respecto de qué inclinación y orientación deben tener objetos como techos, paredes, paneles solares y demás. Si bien este desarrollo representa un producto lucrativo, a la fecha de redacción de este artículo, el software Apphelson está disponible de forma libre y gratuita, sin publicidades ni requisitos de suscripción, entre otras limitaciones.

Referencias

- [1] EGGLY, G. M., MICHELETTO, M., D'AMICO, J. P., AND CROCIONI, S. J. Desarrollo de una aplicación móvil para el cálculo de pulverizaciones agrícolas. *46 JAIIO - Congreso Argentino de AgroInformática 9, 4* (2017), 218–225.
- [2] MARTIN, R. C. *Clean Architecture: A Craftsman's Guide to Software Structure and Design*, 1st ed. Prentice Hall Press, USA, 2017.
- [3] MICHELETTO, M. Agrario: una aplicación multiplataforma para la gestión de tareas agropecuarias. *52 JAIIO - Congreso Argentino de AgroInformática 9, 4* (2023), 97–110.
- [4] MICHELETTO, M., EGGLY, G. M., D'AMICO, J. P., AND CROCIONI, S. J. Desarrollo de una aplicación híbrida para cálculos de siembra. *50 JAIIO - Congreso Argentino de AgroInformática 9, 4* (2021), 189–199.
- [5] OU, C.-J. Optimization of angular setup with proposed $\beta - \eta$ chart for solar energy apparatus through helios orbital analysis. *Energy Conversion and Management 81* (2014), 354–362.

- [6] SALAZAR TRUJILLO, J. H. Calculation of the shadow-penumbra relation and its application on efficient architectural design. *Solar Energy* 110 (2014), 139–150.
- [7] WANG, X., ZHANG, X., ZHU, S., REN, J., CAUSONE, F., YE, Y., JIN, X., ZHOU, X., AND SHI, X. A novel and efficient method for calculating beam shadows on exterior surfaces of buildings in dense urban contexts. *Building and Environment* 229 (2023), 109937.
- [8] ZALAMEA-LEÓN, E., ASTUDILLO-FLORES, M., BARRAGÁN-ESCANDÓN, A., AND PELÁEZ-SAMANIEGO, M. R. Comparative capacities of residential solar thermal systems versus f-chart model predictions and economic potential in an equatorial-latitude country. *Energy Reports* 10 (2023), 2567–2581.