



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Industrial

Diseño de un programa de optimización de una instalación  
fotovoltaica de autoconsumo. Aplicación a comunidades  
energéticas.

Trabajo Fin de Grado

Grado en Ingeniería de la Energía

AUTOR/A: Brucker Ferrando, Andrés Miguel

Tutor/a: Magraner Benedicto, María Teresa

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCOLA TÈCNICA  
SUPERIOR ENGINYERIA  
INDUSTRIAL VALÈNCIA

# Trabajo Fin de Grado

## Escola Tècnica Superior Enginyeria Industrial València

Diseño de un programa de optimización de  
una instalación fotovoltaica de  
autoconsumo.

Aplicación a comunidades energéticas.

**Autor:** Andrés Miguel Brucker Ferrando

**Tutora:** María Teresa Magraner Benedicto

## Resumen

Este trabajo se divide en tres partes distintas: análisis y modelado energético, desarrollo de un programa de dimensionado fotovoltaico y simulación, y predicción de potencia de una planta fotovoltaica utilizando técnicas de Machine Learning.

En la primera parte, se analiza una vivienda unifamiliar modelo en Zaragoza, calculando parámetros relacionados con el control de la demanda energética y comparándolos con los límites establecidos en el Documento Básico de Ahorro de Energía del Código Técnico de la Edificación. Estos parámetros incluyen transmitancia térmica, coeficiente global de transmisión de calor, parámetro de control solar y permeabilidad al aire de los huecos de la envolvente.

En la siguiente etapa, se utiliza el software DesignBuilder para simular el comportamiento energético detallado de la vivienda, considerando aspectos como geometría, materiales, actividad de los ocupantes y consumo de energía. Los resultados muestran un consumo anual significativamente bajo, atribuido a una envolvente térmica eficiente, iluminación LED, ventilación natural y un sistema de climatización basado en aerotermia.

Luego, se presenta un programa en Python que dimensiona una instalación fotovoltaica personalizada considerando perfiles de consumo específicos. Además, se incorporan módulos para evaluar el uso de excedentes para calentar agua y analizar la conveniencia de una comunidad energética.

Finalmente, se aborda la aplicación de técnicas de Machine Learning para predecir la potencia generada por una planta fotovoltaica, utilizando algoritmos como regresión lineal y de árbol de decisión. Aunque se pueden mejorar los resultados con más datos y variables, los algoritmos muestran un rendimiento prometedor en comparación con fórmulas físicas.

En conclusión, este trabajo proporciona un análisis integral y herramientas para optimizar la eficiencia energética, dimensionar instalaciones fotovoltaicas, gestionar excedentes y predecir la potencia de plantas fotovoltaicas, mostrando cómo la combinación de enfoques tecnológicos y analíticos puede mejorar significativamente la sostenibilidad energética y reducir costos para los consumidores.

**Palabras clave:** Fotovoltaica, Python, Inteligencia artificial, Análisis energético

## Resum

Aquest treball es divideix en tres parts diferents: anàlisi i modelatge energètic, desenvolupament d'un programa de dimensionatge fotovoltaic i simulació, i predicció de potència d'una planta fotovoltaica utilitzant tècniques d'Aprenentatge Automàtic.

En la primera part, s'analitza una vivenda unifamiliar model a Saragossa, calculant paràmetres relacionats amb el control de la demanda energètica i comparant-los amb els límits establerts en el Document Bàsic d'Economia d'Energia del Codi Tècnic de l'Edificació. Aquests paràmetres inclouen transmissió tèrmica, coeficient global de transmissió de calor, paràmetre de control solar i permeabilitat a l'aire dels forats de l'envolvent.

En la següent etapa, s'utilitza el programari DesignBuilder per simular el comportament energètic detallat de l'habitatge, considerant aspectes com geometria, materials, activitat dels ocupants i consum d'energia. Els resultats mostren un consum anual significativament baix, atribuït a una envolvent tèrmica eficient, il·luminació LED, ventilació natural i un sistema de climatització basat en aerotèrmia.

Després, es presenta un programa en Python que dimensiona una instal·lació fotovoltaica personalitzada considerant perfils de consum específics. A més, s'incorporen mòduls per avaluar l'ús d'excedents per escalfar aigua i analitzar la conveniència d'una comunitat energètica.

Finalment, s'aborda l'aplicació de tècniques d'Aprenentatge Automàtic per predir la potència generada per una planta fotovoltaica, utilitzant algoritmes com ara la regressió lineal i d'arbre de decisió. Tot i que es poden millorar els resultats amb més dades i variables, els algoritmes mostren un rendiment prometedori en comparació amb fórmules físiques.

En conclusió, aquest treball proporciona una anàlisi integral i eines per optimitzar l'eficiència energètica, dimensionar instal·lacions fotovoltaiques, gestionar excedents i predir la potència de plantes fotovoltaiques, mostrant com la combinació d'enfocaments tecnològics i analítics pot millorar significativament la sostenibilitat energètica i reduir costos per als consumidors.

**Paraules clau:** Fotovoltaica, Python, Intel·ligència artificial, Anàlisi energètic.

## Abstract

This work is divided into three different parts: energy analysis and modeling, development of a photovoltaic sizing and simulation program, and prediction of power output from a photovoltaic plant using Machine Learning techniques.

In the first part, an analysis is conducted on a model single-family home in Zaragoza, calculating parameters related to controlling energy demand and comparing them with the limits set in the Basic Energy Saving Document of the Technical Building Code. These parameters include thermal transmittance, overall heat transmission coefficient, solar control parameter, and air permeability of the building envelope openings.

In the next stage, the DesignBuilder software is used to simulate the detailed energy behavior of the home, considering aspects such as geometry, materials, occupant activity, and energy consumption. The results show a significantly low annual consumption, attributed to an efficient thermal envelope, LED lighting, natural ventilation, and a highly efficient aerothermal-based heating and cooling system.

Next, a Python program is presented that sizes a customized photovoltaic installation considering specific consumption profiles. Modules are included to assess the use of excess energy for water heating and to analyze the feasibility of an energy community.

Finally, the application of Machine Learning techniques to predict the power generated by a photovoltaic plant is discussed, utilizing algorithms such as linear regression and decision trees. Although results can be improved with more data and variables, the algorithms show promising performance compared to explicit formulas.

In conclusion, this work provides a comprehensive analysis and tools to optimize energy efficiency, size photovoltaic installations, manage excess energy, and predict the power of photovoltaic plants, demonstrating how the combination of technological and analytical approaches can significantly enhance energy sustainability and reduce costs for consumers.

**Keywords:** Photovoltaic, Python, Artificial intelligence, Energy analysis.

# Índice

1. Antecedentes .....	8
2. Motivación y justificación.....	8
3. Metodología .....	9
4. Normativa aplicable .....	10
5. Modelado Energético de la Instalación.....	11
5.1. Introducción .....	11
5.2. Comprobaciones del Código Técnico de la Edificación (CTE).....	11
5.2.1. Límite en la transmitancia de cada elemento de la envolvente térmica .....	11
5.2.2. Límite del coeficiente global de transmisión de calor (K) a través de la envolvente térmica .....	12
5.2.3. Control solar de la envolvente térmica .....	15
5.2.4. Permeabilidad al aire de la envolvente térmica.....	16
5.3. Simulación en DesignBuilder .....	16
5.3.1. Diseño.....	16
5.3.2. Cerramientos.....	16
5.3.3. Actividad.....	16
5.3.4. Calefacción, Ventilación y Aire Acondicionado .....	20
5.3.5. Resultados .....	23
6. Instalación Fotovoltaica .....	27
6.1. Introducción .....	27
6.2. Módulo config.py: Función selector_principal.....	28
6.2.1. Propósito .....	28
6.2.2. Conceptos Previos .....	28
6.2.3. Estructura Básica de la Función.....	31
6.2.4. Parámetros de Entrada .....	31
6.2.5. Funciones y Cálculos Realizados .....	31
6.2.6. Observaciones .....	32
6.3. Módulo inverter.py: Función selector_inversor .....	33
6.3.1. Propósito .....	33
6.3.2. Conceptos previos .....	33
6.3.3. Estructura básica de la función .....	36
6.3.4. Parámetros de entrada .....	36
6.3.5. Funciones y cálculos realizados.....	36
6.4. Módulo acs.py .....	37

6.4.1. Propósito .....	37
6.4.2. Funciones .....	38
6.4.3. Gráficos .....	44
6.4.4. Observaciones .....	48
6.5. Módulo <code>electrico.py</code> : Función <code>calcular_S</code> .....	49
6.5.1. Propósito .....	49
6.5.2. Conceptos previos .....	49
6.5.3. Fórmulas empleadas .....	49
6.5.4. Estructura básica de la función .....	50
6.5.5. Parámetros de entrada .....	50
6.5.6. Funciones y cálculos realizados .....	51
6.5.7. Observaciones .....	51
6.6. Módulo <code>modulos_tfg</code> .....	51
6.6.1. Propósito .....	51
6.6.2. Funciones .....	52
6.7. Módulo ' <code>simulador.py</code> ' .....	53
6.7.1. Propósito .....	53
6.7.2. Funciones .....	54
6.8. Módulo <code>modulo_principal</code> .....	57
6.8.1. Propósito .....	57
6.8.2. Cálculos y funciones realizadas .....	57
6.8.4. Resultados .....	60
6.9. Inyección de los Excedentes en la Red Eléctrica .....	65
6.9.1. Modalidades .....	65
6.9.1. Mecanismo de compensación simplificada .....	66
6.10. Estimación del importe ahorrado .....	66
7. Empleo de técnicas de Machine Learning para la predicción de la potencia generada en plantas fotovoltaicas .....	68
7.1. Introducción .....	68
7.2. Algoritmos de regresión .....	68
7.2.1. Regresión lineal .....	68
7.2.2. Regresión polinomial .....	70
7.2.3. Árboles de decisión .....	71
7.3. Métricas de validación .....	72
7.4. Sobreajuste y subajuste .....	72
7.5. Proyecto .....	73

7.6. Resultados .....	80
7.7. Observaciones .....	84
8. Conclusiones.....	85
9. Bibliografía .....	87
<b>CERRAMIENTOS</b> .....	90
<b>ANEXOS</b> .....	96
ANEXO 1. Relación del trabajo con los objetivos de desarrollo sostenible de la agenda 2030 ...	97
ANEXO 2. Figuras.....	99
ANEXO 3. Ganancias solares de las ventanas exteriores .....	102
ANEXO 4. Galería de Imágenes del Modelo de Vivienda en DesignBuilder.....	104
ANEXO 5: Perfiles de Ocupación .....	109
ANEXO 6: Perfiles de Uso de la Iluminación.....	112
ANEXO 7: SCOP en Clima Medio y Cálido – Sistema de Aerotermia de la Marca Baxi .....	114
ANEXO 8: Código Python.....	115
ANEXO 9: Presupuesto .....	144

## 1. Antecedentes

La energía es el motor que impulsa la economía mundial y el bienestar de las sociedades. A lo largo del último siglo, la dependencia de los combustibles fósiles ha llevado a graves problemas medioambientales, incluyendo el calentamiento global y la contaminación atmosférica. Además, los combustibles fósiles son recursos finitos y su agotamiento inevitable ha planteado la necesidad de buscar fuentes de energía alternativas y sostenibles.

En este contexto, han surgido las energías renovables como una solución prometedora. Las fuentes de energía renovable, como la solar, eólica, hidroeléctrica, geotérmica y biomasa, ofrecen una forma sostenible y respetuosa con el medio ambiente de satisfacer la demanda de energía.

En particular, la energía solar fotovoltaica ha experimentado un rápido crecimiento en las últimas décadas debido a su abundancia, disponibilidad y los avances tecnológicos que han reducido sus costos. A pesar de sus ventajas, la energía solar presenta desafíos, como su variabilidad y la necesidad de almacenamiento o balanceo para compensar los períodos de baja producción.

En respuesta a estos desafíos, han surgido las comunidades energéticas. Estas son agrupaciones de hogares, empresas u organizaciones que producen, consumen y comparten energía de manera colectiva. En el ámbito de la energía solar, las comunidades energéticas pueden ayudar a equilibrar la demanda y la oferta, compartir el excedente de energía y reducir los costos al compartir las infraestructuras y los sistemas de almacenamiento. A pesar de la importancia de las comunidades energéticas, en este estudio se analizará su implementación en un contexto más amplio. Se explorarán aspectos como el desplazamiento de la producción de agua caliente sanitaria (ACS), así como la venta de energía a la red.

## 2. Motivación y justificación

Dado este contexto, es evidente la necesidad de explorar y entender mejor el potencial y las implicaciones de las comunidades energéticas. El objetivo principal de este Trabajo Final de Grado es analizar y optimizar el uso de la energía solar fotovoltaica, haciendo uso de todo el conocimiento posible, como los detalles referentes al aspecto constructivo de la vivienda, las variables meteorológicas asociadas a la ubicación, la actividad de los integrantes de la vivienda y su empleo de los equipos contenidos en esta, así como del empleo de datos técnicos relacionados con los equipos asociados a la instalación fotovoltaica, entre otros.

Este estudio es importante por varias razones. En primer lugar, puede ayudar a mejorar la eficiencia y la eficacia del uso de la energía solar, reduciendo así la dependencia de los combustibles fósiles y contribuyendo a la lucha contra el cambio climático. En segundo lugar, puede ayudar a reducir los costos de la energía para los hogares y las empresas, mejorando su bienestar y competitividad. Por último, los hallazgos de este estudio pueden proporcionar información valiosa para los responsables de las políticas, los planificadores de la energía y los inversores sobre las mejores prácticas para el diseño y la implementación de las comunidades energéticas.

### 3. Metodología

Para lograr estos objetivos, se lleva a cabo un estudio de simulación basado en una combinación de técnicas de modelado y análisis. El primer paso es modelar una vivienda utilizando el software DesignBuilder, que es una herramienta de modelado de información de edificios (BIM) utilizada para el análisis de energía en edificios.

Utilizando DesignBuilder, se crea un modelo detallado de la vivienda, incluyendo su orientación, ubicación geográfica, construcción, materiales y sistemas de energía. A continuación, se utiliza este modelo para calcular el consumo de energía de la vivienda a lo largo de un año típico en la ubicación geográfica seleccionada.

El siguiente paso es desarrollar un código en Python para simular la instalación y operación de un sistema de energía solar fotovoltaica en la vivienda. El código incluye varias funciones que dimensionan la instalación de forma que el precio del kilovatio-hora sea mínimo, haciendo uso de datos meteorológicos, de los datos técnicos de los paneles y de la electrónica de potencia asociada y del consumo, entre muchos otros, con el objetivo de minimizar el precio del kilovatio-hora. Por otra parte, incluye módulos que permiten analizar el impacto de la redistribución de la producción de ACS a los momentos de excedente fotovoltaico, comparar los resultados de dimensionar conjuntamente la instalación para un grupo de viviendas frente a hacerlo de forma individual y simular el impacto de la creación de una comunidad energética.

Para analizar la viabilidad de formar una comunidad energética, se generan y manipulan bases de datos de consumo de energía para simular diferentes patrones de consumo entre las viviendas. A partir de estas bases de datos, se identifican los períodos de déficit y excedente de energía, y el impacto de la conformación de una comunidad energética dados los perfiles obtenidos.

Para llevar a cabo una gestión eficiente de la energía y optimizar el rendimiento de los sistemas fotovoltaicos, puede ser muy conveniente contar con una predicción de la generación de energía solar. En este contexto, los algoritmos de aprendizaje automático (Machine Learning, ML) pueden desempeñar un papel importante.

En este estudio, se emplean técnicas de aprendizaje automático para predecir la generación de energía solar en base a datos históricos y variables ambientales. Los algoritmos de aprendizaje automático pueden modelar las relaciones no lineales y las interacciones complejas entre las variables de entrada que afectan a la demanda de energía y la generación de energía solar, lo que permite una predicción más precisa que los métodos tradicionales.

Con el fin de dar a conocer este tipo de aprendizaje, se introducen algunas técnicas sencillas de cara a estudiar su utilidad y se utilizan varios algoritmos de aprendizaje automático, como la regresión lineal, la regresión polinomial y los árboles de decisión, para predecir la generación de energía solar. Se utiliza la validación cruzada para evaluar el rendimiento de los modelos y seleccionar el modelo óptimo. Una vez obtenido un modelo suficientemente preciso, es posible utilizarlo con diversos fines, entre ellos la detección de anomalías, el mantenimiento predictivo y la gestión de los equipos.

En resumen, la integración de las técnicas de aprendizaje automático en la gestión de la energía solar puede mejorar la eficiencia y la confiabilidad del sistema, reducir los costos de operación y mantenimiento, y aumentar la sostenibilidad de la energía solar.

## 4. Normativa aplicable

Documento Básico HE Ahorro de energía, HE 1: "Condiciones para el control de la demanda energética".

Real Decreto 244/2019, de 5 de abril: "Establece las condiciones administrativas, técnicas y económicas del autoconsumo de energía eléctrica".

Reglamento Electrotécnico para Baja Tensión (REBT), aprobado por el Real Decreto 842/2002, de 2 de agosto:

- ITC-BT-40: "Requisitos para instalaciones generadoras aisladas de la red o conectadas a la red a través de un convertidor electrónico".
- ITC-BT-19: "Define las condiciones de seguridad y características para distintos tipos de conductores y cables".

Pliego de Condiciones Técnicas de Instalaciones Conectadas a Red del IDAE: "Guía de recomendaciones y condiciones técnicas para instalaciones conectadas a la red".

## 5. Modelado Energético de la Instalación

### 5.1. Introducción

La vivienda escogida para el proyecto es de tipo unifamiliar. Los planos y la definición de los cerramientos han sido sacados de la 'Guía de aplicación DB HE 2019 - Ejemplos (I)' del Código Técnico de la Edificación (CTE) (Ministerio de Transportes, Movilidad y Agenda Urbana, 2019). Una representación de estos elementos puede encontrarse en la sección "Figuras" y "Cerramientos", respectivamente, situadas al final del documento. El emplazamiento seleccionado para la vivienda se encuentra en Zaragoza, con una latitud de 41.581 y una longitud de -0.919. Esta provincia se sitúa al noreste de España. La zona cuenta con una topografía plana y una densidad poblacional moderada. El clima de la región es mediterráneo, con marcada influencia continental, caracterizado por escasas precipitaciones (320 mm anuales) y temperaturas en promedio moderadas (15,3 °C). La diferencia de temperaturas entre el invierno es notable. En un enero frío, la media se sitúa en los 6,6 °C y la mínima entorno a los -3° C. Por otra parte, en un julio cálido, la temperatura promedio es de 24,9 °C y la máxima llega a los 40 °C. La presencia de vientos fuertes es ocasional en la zona, llegando a superar en ocasiones los 100 Km/h.

### 5.2. Comprobaciones del Código Técnico de la Edificación (CTE)

Estas comprobaciones se realizan para asegurar el cumplimiento de los requisitos del 'Documento Básico de Ahorro de Energía (DB-HE)' del Código Técnico de la Edificación (CTE) (Ministerio de Transportes, Movilidad y Agenda Urbana, 2019). En particular, de los requerimientos del DB-HE1 en el cual se establecen las condiciones para el control de la demanda energética. Esta norma da las directrices para diseñar un edificio de tal manera que se necesite la mínima energía posible para mantener un ambiente confortable, teniendo en cuenta factores como el clima y el uso que se hace del edificio.

#### 5.2.1. Límite en la transmitancia de cada elemento de la envolvente térmica

La transmitancia térmica ( $U$ ) de cada elemento perteneciente a la envolvente térmica no superará el valor límite ( $U_{lim}$ ) de la tabla 3.1.1.a-HE1:

Elemento	Zona climática de invierno					
	$\alpha$	A	B	C	D	E
Muros y suelos en contacto con el aire exterior ( $U_s, U_M$ )	0,8	0,7	0,56	0,49	0,41	0,37
Cubiertas en contacto con el aire exterior ( $U_c$ )	0,55	0,5	0,44	0,4	0,35	0,33
Muros, suelos y cubiertas en contacto con espacios no habitables o con el terreno ( $U_T$ )	0,9	0,8	0,75	0,7	0,65	0,59
Medianerías o particiones interiores pertenecientes a la envolvente térmica ( $U_{MD}$ )						
Huecos (conjunto de marco, vidrio y, en su caso, cajón de persiana) ( $U_H$ )	3,2	2,7	2,3	2,1	1,8	1,8
Puertas con superficie semitransparentes igual o inferior al 50%	5,7					

Tabla 1: Valores límite de transmitancia térmica,  $U_{lim}$  [ $W/m^2K$ ]. Fuente: DB HE1 – Código Técnico de la Edificación.

Dado que el emplazamiento escogido se encuentra entre los 250 y los 300 metros sobre el nivel del mar, la zona climática corresponde a la "D", de acuerdo con la tabla a del Anexo B del 'Documento Básico de Ahorro de Energía (DB-HE)' del Código Técnico de la Edificación (CTE) (Ministerio de Transportes, Movilidad y Agenda Urbana, 2019).

En la siguiente tabla se puede observar en la columna central la transmitancia de cada uno de los elementos pertenecientes a la envolvente térmica, agrupados por tipología, dado que las capas de estos comparten composición y disposición.

Cerramiento	U [W/m <sup>2</sup> - K]	U Lim. [W/m <sup>2</sup> - K]
Suelo	0,19	0,65
Muro fachada	0,214	0,41
Cubierta	0,148	0,35
Hueco puerta	2,2	5,7
Hueco ventana	1,14	1,8
Hueco lucernario	1,29	1,8

Tabla 2: Transmitancia térmica frente a transmitancia térmica límite.

Teniendo los valores límite de transmitancia situados en la columna derecha, se comprueba que la vivienda cumple con el primer requisito.

### 5.2.2. Límite del coeficiente global de transmisión de calor (K) a través de la envolvente térmica

El coeficiente global de transmisión de calor a través de la envolvente térmica (K) del edificio, o parte de este, con uso residencial privado, no superará el valor límite (Klim) obtenido en la tabla 3.1.1.b-HE1.

	Compacidad V/A [m <sup>3</sup> /m <sup>2</sup> ]	Zona climática de invierno					
		$\alpha$	A	B	C	D	E
<b>Edificios nuevos y ampliaciones</b>	V/A ≤ 1	0,67	0,6	0,58	0,53	0,48	0,43
	V/A ≥ 4	0,86	0,8	0,77	0,72	0,67	0,62
<b>Cambios de uso. Reformas en las que se renueve más del 25% de la superficie total de la envolvente térmica del edificio</b>	V/A ≤ 1	1	0,87	0,83	0,73	0,63	0,54
	V/A ≥ 4	1,07	0,94	0,9	0,81	0,7	0,62

Tabla 3: Valor límite Klim [W/m<sup>2</sup>K] para uso residencial privado. Fuente: DB HE1 – Código Técnico de la Edificación.

En primer lugar, se ha calculado la compacidad, la cual expresa la relación entre el volumen encerrado por la envolvente térmica y su superficie, dado que los valores límites para K son más o menos restrictivos también en función de este valor.

Como puede apreciarse en la Tabla 5 la superficie perteneciente a la envolvente térmica resulta en un total de 297,91 m<sup>2</sup>. No obstante, para el cálculo del parámetro en cuestión se computan exclusivamente aquellas superficies en contacto directo con el aire exterior o con el terreno. Dada la existencia de una cámara sanitaria entre el terreno y el suelo de la planta baja, se excluye la superficie de este último, obteniendo un valor de 233,91 m<sup>2</sup>.

El volumen encerrado por la Envolvente Térmica (ET) se ha calculado según la siguiente expresión.

$$V \in ET = V \text{ ortoedro} + V \text{ pirámide cuadrangular}$$

*Ecuación 1: Volumen de un objeto tridimensional compuesto por un ortoedro y una pirámide cuadrangular*

$$V \in ET = (8 \times 8 \times 4,71) + \left(\frac{1}{3} \times 8^2 \times 4\right)$$

$$V \in ET = 386,77 \text{ m}^3$$

La compacidad del modelo resulta en un valor de 1,65 [m<sup>3</sup>/m<sup>2</sup>].

El parámetro K puede calcularse de forma simplificada según la siguiente expresión:

$$K = \frac{\sum_x b_{tr,x} \times [\sum_i A_{x,i} \times U_{x,i} + \sum_k l_{x,k} \times \psi_{x,k} + \sum_j X_{x,j}]}{\sum_x \sum_i b_{tr,x} \times A_{x,i}}$$

*Ecuación 2: Coeficiente global de transmisión de calor (a través de la envolvente térmica de edificio)*

- $B_{tr,x}$ : Factor de ajuste para elementos de la envolvente. Toma un valor de 1 excepto para elementos adyacentes exteriores a la envolvente térmica.
- $A_{x,i}$ : Área de intercambio del elemento de la envolvente térmica evaluado.
- $U_{x,i}$ : Valor de la transmitancia térmica del elemento de la envolvente térmica evaluado.
- $l_{x,k}$ : Longitud del puente térmico evaluado.
- $\Psi_{x,k}$ : Valor de la transmitancia térmica lineal del puente térmico evaluado.
- $X_{x,j}$ : Transmitancia puntual del puente térmico evaluado.

Los cálculos se han resumido en las dos siguientes tablas.

Tipo	Nº	L. unid. [m]	L. total [m]	Transmitancia lineal [W/m-K]	Transferencia térmica [W/K]
Jambas (Huecos fachada P1)	8	2,2	17,6	0,03	0,53
Jambas (Huecos fachada P2)	8	1,5	12	0,03	0,36
Jambas (Lucernarios)	8	1	8	0,03	0,24
Dinteles (Lucernarios)	4	1	4	0,02	0,08
Dinteles y capialzados ventanas	17	1	17	0,22	3,74
Alféizares (Lucernarios)	4	1	4	0,08	0,32
Alféizares (Huecos)	17	1	17	0,51	8,67
Frentes de forjado	4	8	32	0,1	3,2
Cubiertas	4	8	32	0,23	7,36
Esquinas (Bloque)	4	4,71	18,84	0,05	0,94
Esquinas (Cubierta)	4	6,36	25,46	0,05	1,27
<b>Total</b>	-	-	<b>187,90</b>	-	<b>26,71</b>

Tabla 4: Transmitancia lineal y transferencia térmica.

Planta	Cerramiento	O	Nº	S. Ext. [m <sup>2</sup> ]	S. Int. [m <sup>2</sup> ]	S.E.T. [m <sup>2</sup> ]	U [W /m <sup>2</sup> -K]	U x A [W/K]
1	Suelo	-	1	81	64	64	0,19	0
1	Muro exterior	N	1	27	24	17,4	0,22	3,83
1	Muro exterior	S	1	27	24	13	0,22	2,86
1	Muro exterior	E	1	27	24	17,4	0,22	3,83
1	Muro exterior	W	1	27	24	19,6	0,22	4,31
1	Puerta acceso	W	1	2,2	2,2	2,2	2,2	4,84
1	Ventana (Cristal)	N	3	4,95	4,95	4,95	1	4,95
1	Ventana (Cristal)	S	5	8,25	8,25	8,25	1	8,25
1	Ventana (Cristal)	E	3	4,95	4,95	4,95	1	4,95
1	Ventana (Cristal)	W	1	1,65	1,65	1,65	1	1,65
1	Ventana (Marco)	N	3	1,65	1,65	1,65	1,5	2,48
1	Ventana (Marco)	S	5	2,75	2,75	2,75	1,5	4,13
1	Ventana (Marco)	E	3	1,65	1,65	1,65	1,5	2,48
1	Ventana (Marco)	W	1	0,55	0,55	0,55	1,5	0,83
2	Suelo	-	0	0	0	0	0	0
2	Muro exterior	N	1	13,5	12	9,8	0,22	2,16
2	Muro exterior	S	1	13,5	12	7,6	0,22	1,67
2	Muro exterior	E	1	13,5	12	12	0,22	2,64
2	Muro exterior	W	1	13,5	12	9,8	0,22	2,16
2	Ventana (Cristal)	N	1	1,65	1,65	1,65	1	1,65
2	Ventana (Cristal)	S	2	3,3	3,3	3,3	1	3,3
2	Ventana (Cristal)	E	0	0	0	0	0	0
2	Ventana (Cristal)	W	1	1,65	1,65	1,65	1	1,65

2	Ventana (Marco)	N	1	0,55	0,55	0,55	1,5	0,825
2	Ventana (Marco)	S	2	1,1	1,1	1,1	1,5	1,65
2	Ventana (Marco)	E	0	0	0	0	0	0
2	Ventana (Marco)	W	1	0,55	0,55	0,55	1,5	0,83
2	Cubierta	N	1	28,64	22,63	21,03	0,17	3,57
2	Cubierta	S	1	28,64	22,63	20,83	0,17	3,54
2	Cubierta	E	1	28,64	22,63	22,23	0,17	3,78
2	Cubierta	W	1	28,64	22,63	21,83	0,17	3,71
2	Lucernario (Cristal)	N	1	0,80	0,80	0,80	1	0,8
2	Lucernario (Cristal)	S	2	1,60	1,60	1,60	1	1,6
2	Lucernario (Cristal)	E	0	0	0	0	0	0
2	Lucernario (Cristal)	W	1	0,80	0,80	0,80	1	0,8
2	Lucernario (Marco)	N	1	0,20	0,20	0,20	2,4	0,48
2	Lucernario (Marco)	S	2	0,40	0,40	0,40	2,4	0,96
2	Lucernario (Marco)	E	0	0,00	0,00	0,00	0	0
2	Lucernario (Marco)	W	1	0,20	0,20	0,20	2,4	0,48
-	<b>Total</b>	-	-	<b>398,95</b>	<b>339,91</b>	<b>297,91</b>	-	<b>87,62</b>

Tabla 5: Superficies y transmitancias de los cerramientos.

$$K = \frac{26,71 + 87,62}{233,91} = 0,489$$

Interpolando linealmente entre 0,48 (Compacidad = 1) y 0,67 (Compacidad = 4):

$$K_{lim} = 0,521 > 0,489 = K$$

### 5.2.3. Control solar de la envolvente térmica

El parámetro de control solar no superará el valor límite de la tabla 3.1.2-HE1.

Uso	$q_{sol,jul}$
Residencial privado	2
Otros usos	4

Tabla 6: Valor límite del parámetro de control solar,  $q$  [ $kWh/m^2 \cdot mes$ ]. Fuente: DB HE1 – Código Técnico de la Edificación.

Este parámetro se obtiene como cociente entre las ganancias solares durante el mes de julio a través de los huecos con las protecciones solares móviles en funcionamiento y, la superficie útil de los espacios pertenecientes a la envolvente.

Pese a no haber llegado aún a la parte de simulación energética, se ha hecho uso de las ganancias solares obtenidas en los resultados de esta, y que toman un valor de 114,8 kWh. Estos resultados pueden consultarse en el Anexo 1: Ganancias Solares de las Ventanas Exteriores.

$$q_{sol} = \frac{114,8}{128} = 0,897 < 2$$

### 5.2.2. Permeabilidad al aire de la envolvente térmica

La permeabilidad al aire ( $Q_{100}$ ) de los huecos que pertenezcan a la envolvente térmica no superará el valor límite de la tabla 3.1.3.a-HE1:

	Zona climática de invierno					
	$\alpha$	A	B	C	D	E
<b>Permeabilidad al aire de huecos (<math>Q_{100,lim}</math>)*</b>	$\geq 27$	$\geq 27$	$\geq 27$	$\geq 9$	$\geq 9$	$\geq 9$

Tabla 7: Valor límite de permeabilidad al aire de huecos en la envolvente térmica,  $Q_{100,lim}$  [ $m^3/h \cdot m^2$ ]. Fuente: DB HE1 – Código Técnico de la Edificación.

Atendiendo a la norma 'UNE-EN 12207:2017. Ventanas y puertas. Permeabilidad al aire. Clasificación', los huecos pueden clasificarse en función de la cantidad de aire que los atraviesa estando las ventanas cerradas, debido a un diferencial de presión.

Tal y como puede consultarse en la sección “Cerramientos” incluida al final del documento, todos ellos pertenecen a una clase 3, teniendo por tanto una permeabilidad menor o igual que 9 y cumpliendo con la exigencia.

## 5.3. Simulación en DesignBuilder

Haciendo uso de la definición de sus creadores, DesignBuilder es un software especializado en simulación ambiental y energética de edificios. Está concebido para facilitar los procesos de simulación y ofrece herramientas para el diseño, la consultoría y la certificación energética. En las siguientes subsecciones se introducirán y describirán los parámetros más relevantes de cara a una simulación energética.

### 5.3.1. Diseño

Los detalles concernientes a la modelación de la vivienda en el aspecto geométrico se pueden encontrar en el Anexo 2: Galería de Imágenes del Modelo de Vivienda en DesignBuilder.

### 5.3.2. Cerramientos

Todos los datos técnicos relativos a los cerramientos se han introducido directamente en el programa, conforme a la información técnica disponible mencionada en la introducción.

### 5.3.3. Actividad

#### 5.3.3.1. Ocupación

La ocupación queda definida mediante la densidad de ocupación y periodos de ocupación. La densidad de ocupación ha sido definida para cada una de las zonas de forma independiente y el

producto de ésta por la superficie de la zona arroja como resultado el número de personas que se prevé van a utilizar la zona simultáneamente.

Planta	Zona	Superficie [m <sup>2</sup> ]	Densidad de ocupación [personas / m <sup>2</sup> ]	Personas [Nº]
1	Aseo	4,1	0,2463	1
1	Dormitorio principal	17,5	0,1143	2
1	Zona de estar y cocina	41,3	0,0969	4
2	Aseos	4,2	0,241	1
2	Dormitorios	13,6	0,0736	1
2	Zona común	24,3	0,0824	2

Tabla 8: Superficie, densidad de ocupación y personas coincidentes, por zona

De la misma manera, los periodos de ocupación han sido definidos para cada una de las zonas, en algunos casos, distinguiendo entre laborables y no laborables. Estos pueden ser consultados en el Anexo 3: Perfiles de Ocupación.

Para el resto de los parámetros relacionados con la ocupación; tasa metabólica, vestimenta, ponderación de temperatura radiante para confort y velocidad del aire se han tomado los valores por defecto proporcionados por DesignBuilder.

### 5.3.3.2. ACS

Toma un valor de 112 litros por día, dado que según el Anexo F del 'Documento Básico de Ahorro de Energía (DB-HE)' del Código Técnico de la Edificación (CTE), la demanda de referencia diaria de ACS a tener en cuenta para espacios residenciales es de 28 litros por persona (Ministerio de Transportes, Movilidad y Agenda Urbana, 2019). Por otra parte, según la tabla C del Anexo D del 'Documento Básico de Ahorro de Energía (DB-HE)' del CTE, el perfil de uso será el siguiente (Ministerio de Transportes, Movilidad y Agenda Urbana, 2019):

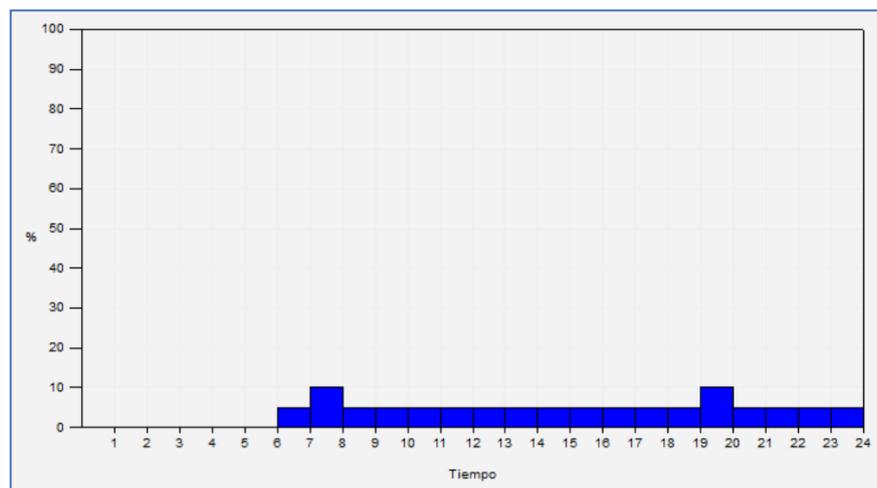


Figura 1: Perfil de uso de ACS

### 5.3.3.3. Control ambiental

#### 1.3.3.3.1 Condiciones operacionales y perfiles de uso

De acuerdo con la tabla a del Anexo D del 'Documento Básico de Ahorro de Energía (DB-HE)' del Código Técnico de la Edificación (CTE), las condiciones operacionales y los perfiles de uso a efectos de cálculo de demanda energética serán los siguientes:

Consigna	Primaria (°C)	Secundaria (°C)
Calefacción	20	17
Refrigeración	25	27

Tabla 9: Temperaturas de consigna

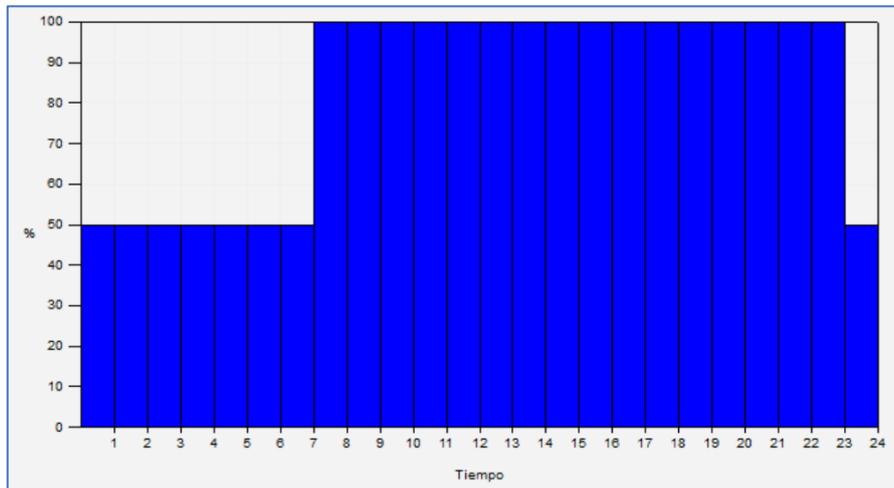


Figura 2: Perfil de uso del sistema de calefacción

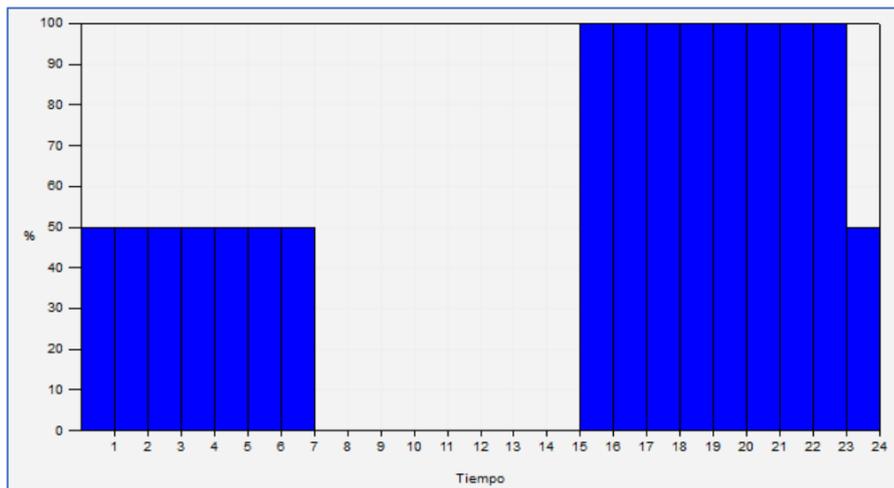


Figura 3: Perfil de uso del sistema de refrigeración

Mediante la consigna primaria se fija el valor que se va a tratar de mantenerse durante el día o periodos de ocupación, mientras que, mediante la secundaria, el valor que va a tratar de mantenerse durante la noche o periodos de desocupación.

Para los perfiles de uso, un valor del 100% equivale a la consigna primaria mientras que, un valor del 50%, a la secundaria.

Las condiciones operacionales para el sistema de calefacción se fijan para los meses que van desde octubre hasta mayo, mientras que, para el sistema de refrigeración, los meses que van de junio a septiembre.

#### 1.3.3.3.2. Ventilación natural

En determinados periodos, es posible utilizar la ventilación natural como recurso de calentamiento o enfriamiento pasivo. De cara a la simulación energética del edificio, se ha establecido un valor de 20 °C como temperatura interior mínima y un valor de 27°C como temperatura exterior máxima. Esto significa, que la ventilación natural puede activarse si la temperatura interior es superior a 20 °C (Consigna de temperatura primaria para calefacción) y si la temperatura exterior es inferior a 27 °C (Consigna de temperatura secundaria para refrigeración). Por último, se define un  $\Delta T$  de 2 °C, es decir, debe existir una diferencia de dos grados centígrados entre interior y exterior para hacer uso de la ventilación natural como recurso.

#### 1.3.3.3.3. Equipos

Se define de acuerdo con el Apartado 8 del documento 'Consumos del Sector Residencial en España' proporcionado por el Instituto para la Diversificación y Ahorro de la Energía (IDAE), en el cual se recoge el consumo medio anual de electrodomésticos por hogar (IDAE, 2011):

Electrodoméstico	Consumo anual [kWh]
Frigorífico	662
Congelador	563
Lavadora	255
Horno	231
Ordenador	172
Televisión	263
Standby	231
Resto	76
<b>Total</b>	<b>2.453</b>

Tabla 10: Consumo anual por electrodoméstico

Se ha introducido en forma de densidad de potencia [ $W/m^2$ ], a nivel de edificio y tomando un valor de 2. Dada la dificultad para predecir en qué momento se va a producir cada consumo se ha utilizado una programación "ON 24/7" es decir, el consumo eléctrico se ha repartido de forma lineal a lo largo del tiempo.

El consumo energético previsto para la cocina, alimentada con gas natural, es de 1.217 kWh/año, tomando como referencia el 'SPAHOUSEC II: Análisis estadístico del consumo de gas natural en las viviendas principales con calefacción individual' proporcionado por el IDAE (IDAE, 2019). Este consumo se ha introducido a nivel de zona y, de igual forma, repartido de forma lineal.

#### 1.3.3.3.4. Iluminación

Atendiendo a los niveles de iluminación recomendables en función del tipo de actividad prevista para cada espacio, se han definido a nivel de zona y como densidad de potencia, los valores reflejados en la tabla de bajo. Tomar en cuenta que, la iluminancia para la zona “Zona de estar y cocina” equivale al promedio de los niveles recomendados para una cocina y una sala de estar.

Planta	Zona	Iluminancia [lux]	Densidad de potencia [W/m <sup>2</sup> ]
1	Aseo	200	1,25
1	Dormitorio principal	150	0,9375
1	Zona de estar y cocina	200	1,25
2	Aseos	200	1,25
2	Dormitorios	150	0,9375
2	Zona común	150	0,9375

Tabla 11: Iluminancia y densidad de potencia

La iluminación específica, como puede ser la requerida en una encimera, el espejo de un baño o la de una lámpara de lectura se ha fijado a nivel de edificio con un valor de 3,125 W/m<sup>2</sup>.

Tómese en cuenta que, para que el apartado de iluminación quede completamente definido, es necesario proporcionar información sobre los perfiles de uso. Estos han sido introducidos mediante una programación “7/12” y pueden encontrarse el Anexo 4: Perfiles de Uso de la Iluminación

#### 5.3.4. Calefacción, Ventilación y Aire Acondicionado

##### 5.3.4.1. Introducción

DesignBuilder ofrece dos formas diferentes de simular las instalaciones de climatización, HVAC simple y HVAC detallado. En HVAC detallado se modelizan cada uno de los equipos que forman parte del sistema de climatización (Caldera, radiadores, climatizadora etc.) con sus propiedades e interconexiones. Por otra parte, el módulo de HVAC simple es un módulo de simulación de instalaciones simplificado, que se basa en un equipo que tiene EnergyPlus que es el equipo de cargas ideales. Es un sistema ideal cuyo funcionamiento lo que hace es una transformación psicométrica, una aportación o extracción de energía al/del aire y eso se contabiliza como un aporte térmico y asignándole un combustible a ese aporte se simula cualquier sistema de climatización.

##### 5.3.4.2. Aerotermia: Análisis y Aplicación

La energía aerotérmica es el calor almacenado en el aire ambiente. Existen sistemas de climatización, fundamentados en bombas de calor, que extraen esta energía para transferirla a otro espacio. Según el medio al que se transfiera el calor extraído, estos sistemas se pueden clasificar en aire-aire y aire-agua. Estos sistemas son altamente eficientes y son considerados como fuente de energía renovable de acuerdo con la Directiva 2009/28/CE del Parlamento Europeo (Parlamento Europeo y Consejo, 2009)

A continuación, se presentan las ventajas y desventajas que conlleva su instalación y empleo:

- Ventajas:
  - Eficiencia energética: Muy eficientes en términos energéticos. Pueden generar varias unidades de energía térmica por unidad de energía eléctrica consumida.
  - Sistemas reversibles: Pueden emplearse tanto para calefacción como para refrigeración.
  - Renovable y respetuosa con el medio ambiente: Al utilizar el calor del aire exterior (interior en el caso de refrigeración), se está utilizando una fuente de energía renovable. Aparte, no producen emisiones directas de gases de efecto invernadero.
  
- Desventajas:
  - Inversión inicial elevada: Este tipo de sistemas requieren de una inversión inicial elevada, lo cual puede suponer un desafío a la hora de decidirse por ellos. La instalación de estos sistemas implica un coste inicial elevado, lo que puede representar un desafío al considerar su implementación. Según el "Estudio comparativo de las diferentes soluciones de instalación térmica para una vivienda unifamiliar" (López Landa, dirigido por Carcedo Haya, 2022), el precio promedio de la instalación de un sistema aerotérmico se sitúa en 9.560 €.
  - Dependencia de la electricidad: Es un aspecto importante a tener en cuenta, especialmente en climas fríos donde un fallo del suministro de electricidad pueda suponer un riesgo para las personas. No obstante, existen maneras de mejorar la fiabilidad del sistema, como su hibridación con sistemas independientes de generación de energía, tal y como puede ser una instalación fotovoltaica como la estudiada durante este proyecto.
  - Necesidad de espacio: Para su instalación, los sistemas aerotérmicos requieren un espacio adecuado que permita colocar la unidad exterior de la bomba de calor. Este requerimiento puede ser un desafío en viviendas o edificios con limitaciones de espacio.

En la simulación ejecutada en DesignBuilder, se ha adoptado un Coeficiente de Rendimiento Estacional (SCOP) de 4.5, utilizando como referencia el sistema de aerotermia Platinum BC iPlus V200 Smart de 4.5 KW de la marca Baxi. Los datos técnicos correspondientes se encuentran disponibles en el Anexo 5: SCOP en Clima Medio y Cálido – Sistema de Aerotermia de la Marca Baxi.

#### *5.3.4.3. Diseño de calefacción*

Existe la posibilidad de que el programa realice el cálculo en función de la temperatura del aire o bien de la temperatura operativa. La temperatura operativa es el promedio entre la temperatura del aire y la temperatura radiante media. Se opta por hacer el cálculo en función de la temperatura operativa dado que el nivel de confort está más relacionado ésta; aparte de la transmisión de calor por convección hacia o desde el aire, existe una transmisión de calor desde o hacia nuestro entorno.

Por último y antes lanzar el módulo de diseño, se ha introducido un coeficiente de seguridad de 1.25. Mediante este se sobredimensionará posteriormente la instalación en un 25% a fin de que los equipos cuenten con la capacidad – potencia necesaria, de hacer frente a condiciones de operación más adversas.

El cálculo del diseño de calefacción es un cálculo que se basa en los estándares de la Sociedad Americana de Ingenieros de Calefacción, Refrigeración y Aire Acondicionado (ASHRAE). Este cálculo se realiza en estado estacionario, suponiendo una temperatura exterior constante a lo largo del día y calculando las pérdidas que tenemos frente a ella. Por otra parte, no se toman en cuenta cargas internas de ningún tipo ni radiación solar ni se toma en cuenta ninguna de las programaciones expuestas anteriormente. Ello resulta en el siguiente gráfico, en el cual podemos observar que todo son pérdidas, excepto el aporte del sistema de calefacción. Este aporte es el resultado del producto entre la pérdida total y el coeficiente de seguridad

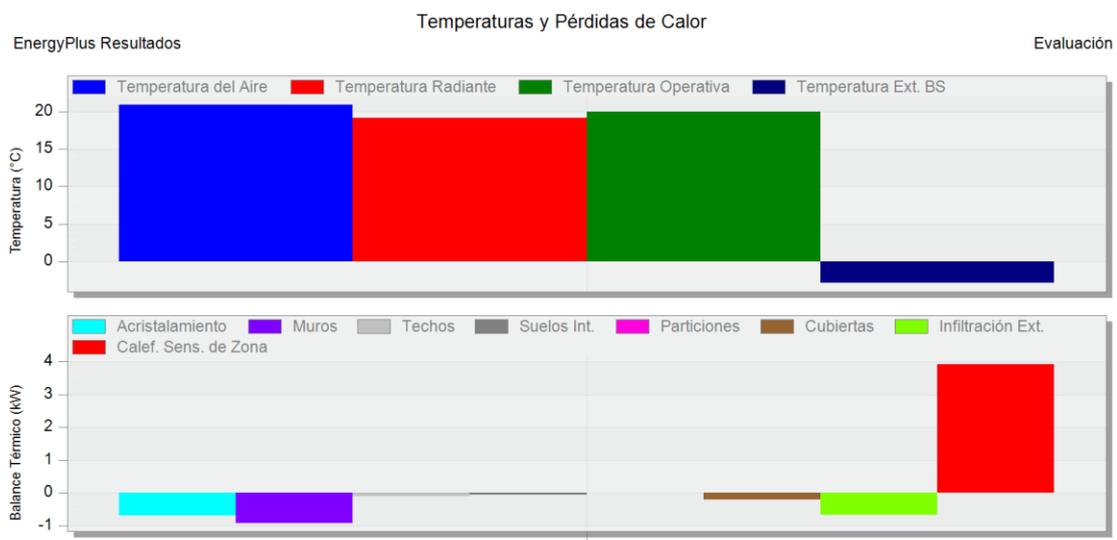


Figura 4: Temperatura y pérdidas de calor

#### 5.3.4.4. Diseño de refrigeración

Al contrario que en el diseño de calefacción, para el diseño de refrigeración, el cálculo se hace en régimen dinámico, tal y como se puede observar en el gráfico de bajo. En este caso, sí se tienen en cuenta tanto las cargas internas como la radiación solar. Para el caso de la refrigeración es necesario definir un día de diseño. Lo habitual y por lo que se ha optado en este proyecto es el día 15 de los meses de verano (Junio – Septiembre).

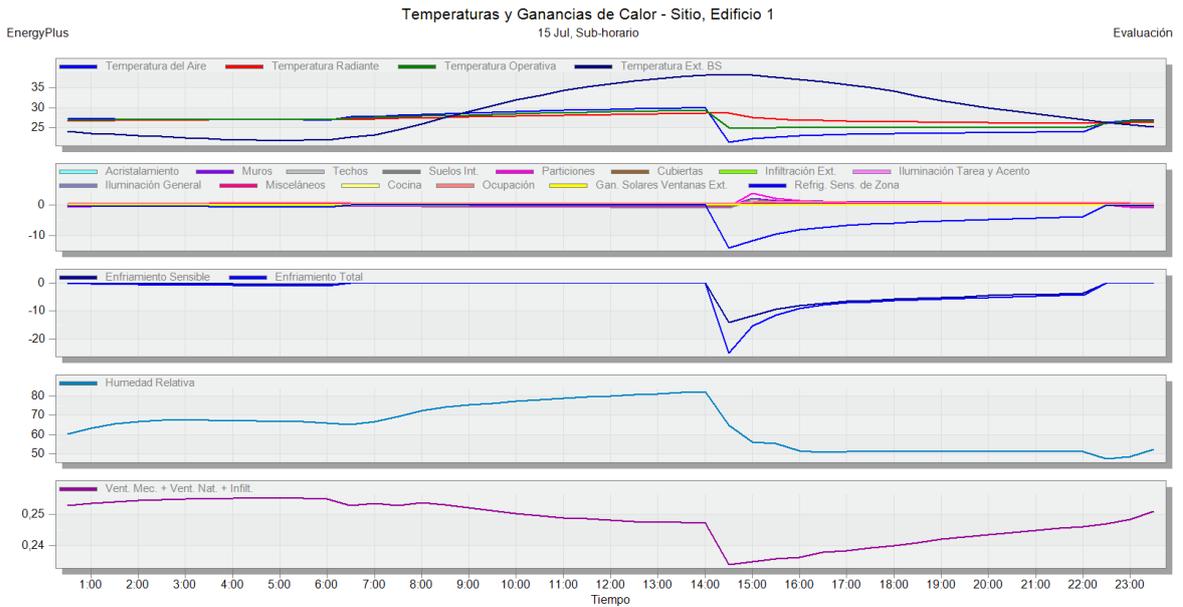


Figura 5: Temperatura y ganancias de calor

### 5.3.5. Resultados

#### 5.3.5.1. Cargas internas y solares

En los siguientes dos gráficos se obtienen resultados sobre las cargas internas, solares y consumo energético necesario por parte de los equipos de climatización para vencerlas, para el periodo completo de simulación.

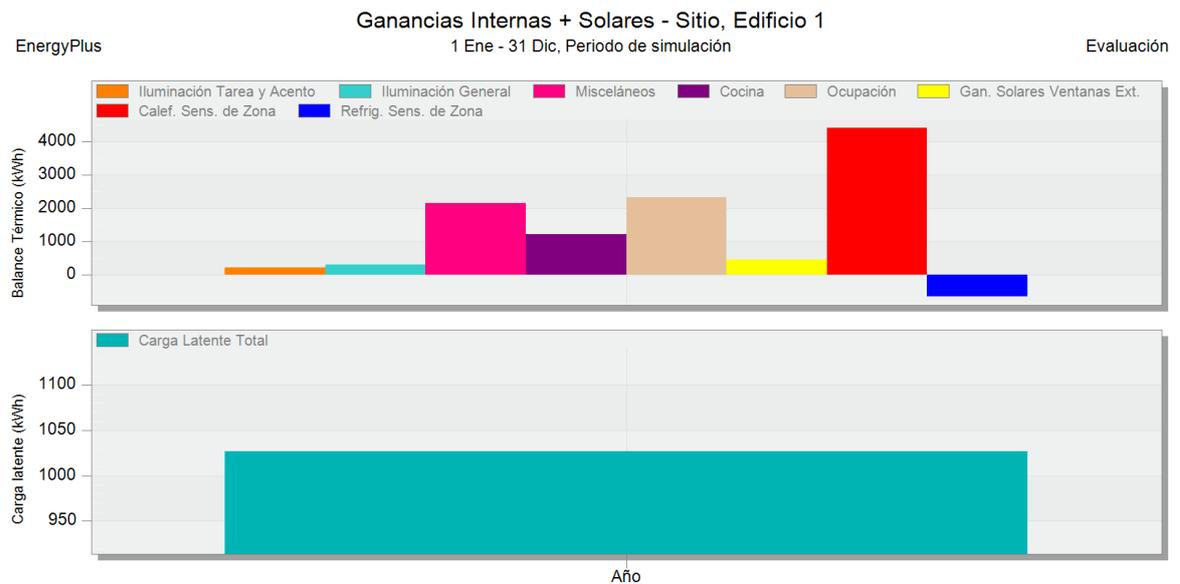


Figura 6: Ganancias internas y solares

### 5.3.5.2. Elementos constructivos y ventilación

En el gráfico dispuesto en la parte inferior es posible apreciar una tasa de renovación constante durante los meses de invierno, correspondiente sobre todo a las infiltraciones mientras que, para los meses que van desde marzo hasta octubre, la tasa de renovación se cuasi quintuplica. Esto es debido al aprovechamiento de la ventilación como método de acondicionamiento de los espacios.

Por otra parte, en el gráfico superior es posible apreciar cómo las cargas de frío debidas a infiltraciones exteriores prácticamente se vuelven cero a lo largo del verano, debido al aumento de la temperatura exterior, mientras que aumentan conforme se acercan épocas más frías. En lo que a la ventilación natural se refiere, por el contrario, las cargas de frío quedan limitadas durante la época más fría, debido a los controles establecidos, mientras que aumentan de forma pronunciada conforme es posible hacer uso de la ventilación natural como sistema de refrigeración pasivo.

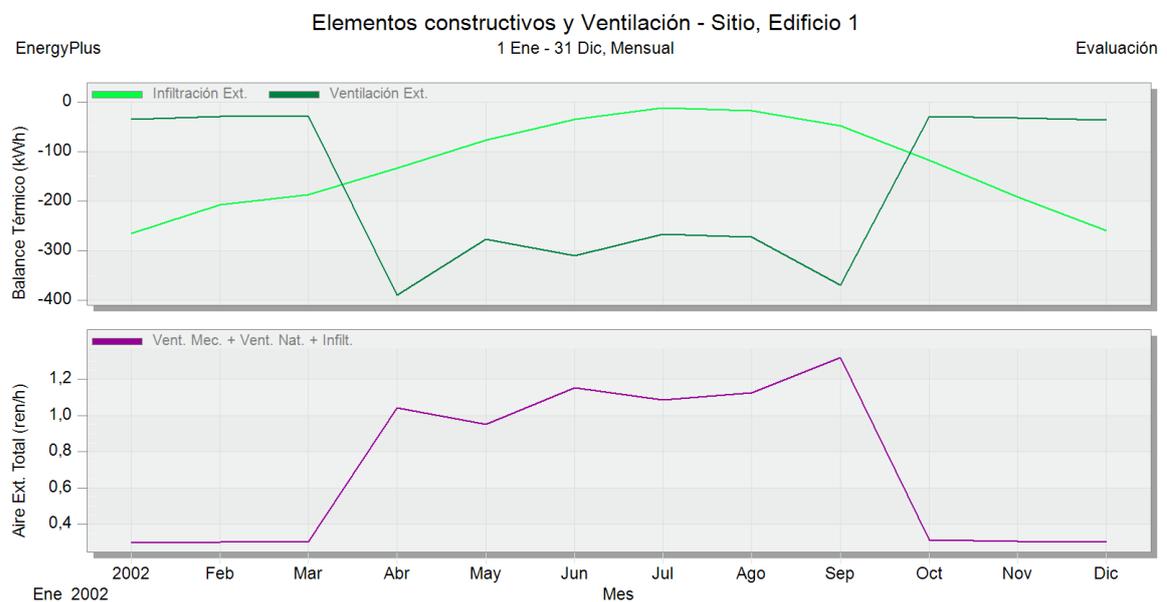


Figura 7: Elementos constructivos y ventilación

### 5.3.5.3. Confort

En el siguiente gráfico, es observable cómo las temperaturas consiguen ser mantenidas dentro de consigna gracias a los equipos de climatización. Se puede apreciar pues, cómo en los momentos más fríos la temperatura operativa se mantiene en el entorno de los 17 °C mientras que en los momentos más cálidos, rara vez se sobrepasan los 25 °C (Pese a tener una consigna secundaria de 27 °C).

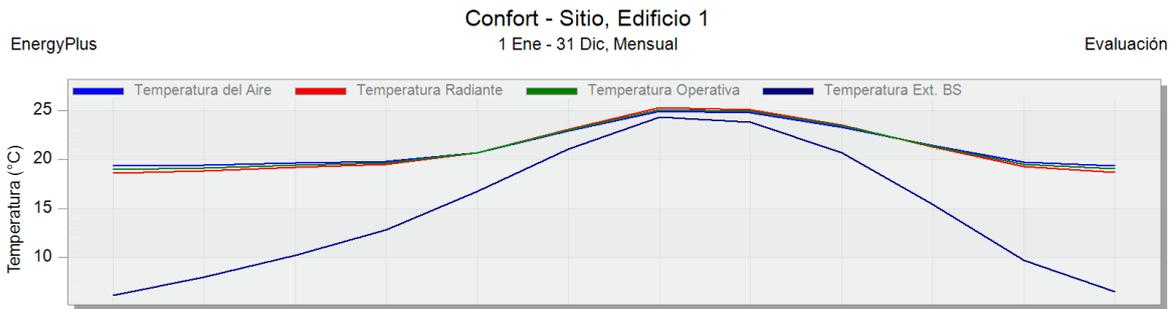


Figura 8: Confort

#### 1.3.5.4. Consumos

En el gráfico de abajo es posible observar los consumos desglosados para el periodo de simulación completo. El consumo eléctrico total para este periodo es de 4.466,921 kWh mientras que el consumo de gas, en términos de energía es de 1.217,272 kWh. (Dato introducido de forma manual. Consultar el apartado “Equipos” del presente capítulo para más información al respecto)

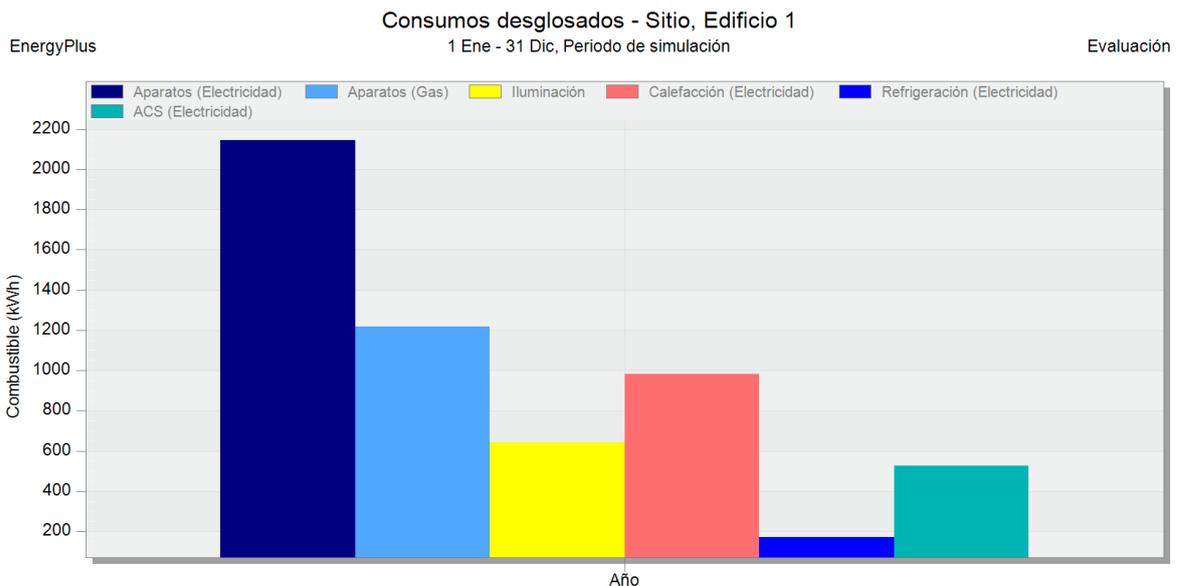
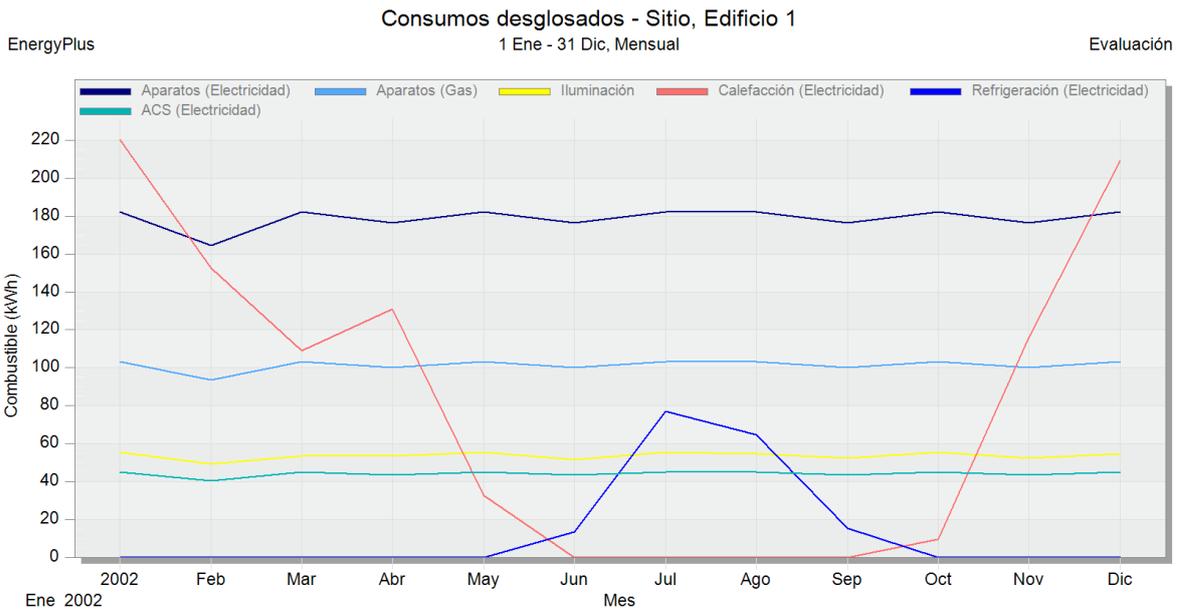


Figura 9: Consumos desglosados – Total anual

Por otra parte, es posible ver la distribución temporal de los consumos. Lo más relevante del siguiente gráfico es la variación temporal de los consumos asociados a los equipos de frío y calor, donde se puede observar el pico de calefacción hacia finales del mes de diciembre y el de climatización hacia finales de junio. Los demás consumos, tal y como comentado anteriormente, adoptan una distribución temporal prácticamente estática, conforme a la programación aplicada, dada la elevada dificultad para predecir los perfiles de uso y la escasa influencia de una distribución más ajustada a la realidad, de cara a una simulación energética.



*Figura 10: Consumos desglosados por mes*

El propósito principal de este capítulo es el de verificar el cumplimiento de los requisitos mínimos establecidos en el Código Técnico de la Edificación. Este proceso se orienta hacia la consecuente modelización y comprensión del comportamiento energético de la vivienda. Es pertinente destacar, que además de esta evaluación, existe la posibilidad de introducir mejoras en la envolvente térmica de la vivienda, mejorando su eficiencia energética y resultando en una disminución de los consumos asociados al sistema de climatización.

Por otra parte, los datos precisos relacionados con el consumo energético adquiridos durante esta fase cumplen una función fundamental como datos de entrada para el diseño de la instalación solar fotovoltaica. Este diseño, se abordará en detalle en el próximo capítulo de este estudio.

## 6. Instalación Fotovoltaica

### 6.1. Introducción

En este capítulo, se presenta un programa desarrollado en Python que se utiliza para dimensionar instalaciones fotovoltaicas y simular distintos escenarios de uso y producción de energía. Durante su ejecución, el programa hace uso de diferentes bases de datos cuidadosamente seleccionadas y compiladas de cara a obtener unos resultados lo más precisos posibles.

Entre las bases de datos utilizadas, se incluyen diversos archivos climáticos del Photovoltaic Geographical Information System (PVGIS), datos técnicos sobre diferentes módulos e inversores fotovoltaicos, perfiles de consumo de energía de la simulación realizada mediante DesignBuilder y una base de datos que contiene las intensidades admisibles por el cobre con sus respectivos factores de corrección de temperatura según el Reglamento Electrotécnico de Baja Tensión.

A continuación, se presenta una tabla que proporciona una mejor comprensión de las bases de datos utilizada. En ella, se especifican los nombres de las bases de datos, el número de entradas y columnas que contienen y una breve descripción de su contenido.

Nombre	Número de Entradas	Número de Columnas	Descripción
database_deg_min_30	17520	4	Datos meteorológicos con ángulo de inclinación ( $\alpha$ ) de $-30^\circ$
database_deg_min_15	17520	4	Datos meteorológicos con ángulo de inclinación ( $\alpha$ ) de $-15^\circ$
database_deg_0	17520	4	Datos meteorológicos con ángulo de inclinación ( $\alpha$ ) de $0^\circ$
database_deg_15	17520	4	Datos meteorológicos con ángulo de inclinación ( $\alpha$ ) de $15^\circ$
database_deg_30	17520	4	Datos meteorológicos con ángulo de inclinación ( $\alpha$ ) de $30^\circ$
consumo	17520	7	Datos sobre los consumos eléctricos de la vivienda
database_modulos	114	22	Base de datos técnica sobre los diferentes módulos fotovoltaicos
database_inversores	67	17	Base de datos técnica sobre los diferentes inversores fotovoltaicos
i_admisibles_cobre	No Aplica	No Aplica	Datos sobre las intensidades admisibles del cobre y factores de corrección para temperaturas diferentes a $30^\circ\text{C}$

Tabla 12: Descripción de las bases de datos

A continuación, se presenta una tabla que resume los módulos del programa. Esta tabla detalla el

nombre de cada módulo, el número de funciones que contiene y función principal. Esta información ayudará a entender mejor la estructura del programa y el papel que juega cada módulo.

Nombre	Número de Funciones	Función Principal
principal	No aplica	Módulo principal que coordina las funciones y módulos subyacentes
config	1	Realiza el cálculo del número de placas y determina la orientación óptima
inversor	1	Selecciona un inversor adecuado y establece las conexiones serie-paralelo óptimas
acs	4	Simula la redistribución de la energía asociada a la producción de ACS
electrico	3	Encargado del cálculo de las secciones necesarias
modulos_tfg	7	Contiene varias funciones de apoyo y cálculos complementarios
simulador	2	Simula la compartición de energía entre las viviendas conformantes de una eventual comunidad energética

*Tabla 13: Descripción de los módulos*

El código del programa ha sido comentado y puede encontrarse en su totalidad en el Anexo 7: Código Python. A lo largo del presente capítulo, se hará una exposición detallada de cada una de las funciones que integran los diversos módulos de los que se compone el programa para que, junto con el código del anexo, sea posible comprender su utilidad y funcionamiento.

## 6.2. Módulo config.py: Función selector\_principal

### 6.2.1. Propósito

El propósito de esta función es el de calcular el número de placas y orientación que hacen mínimo el coste de la energía durante la vida útil de la instalación.

### 6.2.2. Conceptos Previos

#### 6.2.2.1. Influencia de la Irradiancia y la Temperatura en el Rendimiento de las Celdas Fotovoltaicas

La potencia entregada por una celda fotovoltaica concreta depende fundamentalmente de la cantidad de energía radiante (Irradiancia) recibida en un momento determinado. Esto es debido a que su funcionamiento se basa en el llamado efecto fotoeléctrico, por el cual algunos materiales tienen la propiedad de absorber fotones y liberar electrones. Cuando alguno de estos electrones libres es capturado, resulta una corriente eléctrica que puede ser utilizada como electricidad.

Otro factor a tener en cuenta es la temperatura de la celda. Cuando esta aumenta, aumenta la energía térmica de los electrones en el material semiconductor, lo que aumenta la probabilidad de que se recombinen entre sí antes de que puedan ser recolectados y conducidos fuera de la placa. Aparte, un aumento de temperatura provoca un aumento en la resistencia interna de la placa.

A continuación, se pueden observar las curvas I-V correspondientes a un modelo comercial de un fabricante de placas fotovoltaicas.

En este primer gráfico, puede apreciarse que la corriente generada es muy sensible ante fluctuaciones en la irradiancia, mientras que la tensión prácticamente no se ve afectada.

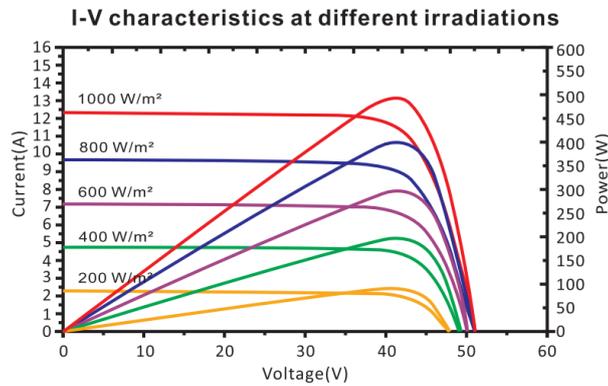


Figura 11: Intensidad frente a Voltaje para diferentes niveles de Irradiancia

En este segundo gráfico puede apreciarse la influencia de la temperatura sobre ambas componentes. En este caso, la tensión tiene una moderada dependencia negativa con la temperatura, mientras que es la corriente la que prácticamente no se ve afectada (Ligera dependencia positiva).

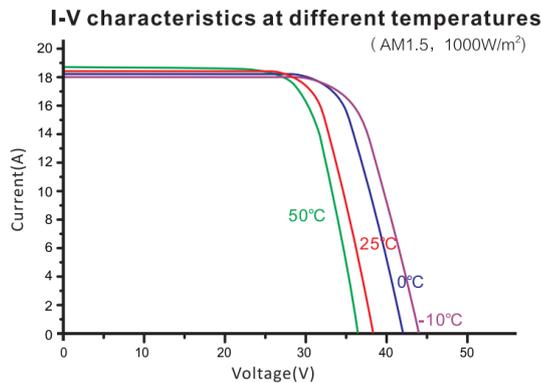


Figura 12: Intensidad frente a Voltaje para diferentes valores de Temperatura

#### 6.2.2.2. Condiciones de Ensayo

Dada la necesidad de establecer una comparación entre los distintos modelos de placas fotovoltaicas, se establecen las siguientes condiciones de prueba, mediante las cuales se determinan las características del panel.

- STC (Standard Test Conditions): Condiciones estándar en la industria
- NOTC (Normal Operating Temperature Conditions): Condiciones más ajustadas a la realidad en la cual van a funcionar las placas.

Condiciones	STC	NOTC
Temperatura de celda / Temperatura del aire (°C)	25	20
Irradiancia (W/m <sup>2</sup> )	1000	800
Masa de aire	1.5	1.5
Velocidad de viento (m/s)	-	1

Tabla 14: Condiciones de Ensayo STC y NOTC

#### 6.2.2.3. Parámetros más Relevantes

- Potencia nominal (W): Potencia entregada por el panel
- Voltaje en circuito abierto ( $V_{oc}$ ): Diferencia de potencial entre los bornes de la placa, cuando no están unidos por ningún medio físico, es decir, cuando no hay flujo de corriente.
- Corriente de cortocircuito ( $I_{sc}$ ): Corriente generada con los bornes del panel cortocircuitados.
- Voltaje en el punto de máxima potencia ( $V_{mpp}$ )
- Corriente en el punto de máxima potencia ( $I_{mpp}$ )
- Eficiencia (%): Indica la relación entre la energía recibida y la entregada. En los modelos comerciales actuales toma valores entorno al 20 %.

#### 6.2.2.4. Fórmulas Empleadas

La siguiente fórmula, relaciona la temperatura de celda con la temperatura ambiente y la irradiancia, a través del parámetro de NOTC.

$$T_c (^{\circ}C) = T_{amb} (^{\circ}C) + G \left( \frac{W}{m^2} \right) \times \frac{NOTC - 20 (^{\circ}C)}{G_{NOTC}} \left( \frac{W}{m^2} \right)$$

Ecuación 3: Corrección de temperatura para radiación solar

- $T_c$ : Temperatura de celda.
- $T_{amb}$ : Temperatura ambiente.
- G: Irradiancia incidente en el panel solar.
- NOTC: Temperatura de la celda fotovoltaica en condiciones NOTC. Es un dato técnico del módulo fotovoltaico.

- $G_{NOTC}$ : Irradiancia de las condiciones NOTC.

Esta fórmula, por su parte, relaciona la potencia entregada por la placa con la irradiancia y la temperatura de celda, a través de la potencia nominal del módulo en condiciones de prueba STC y el coeficiente de temperatura para potencia.

$$P(kW) = G \left( \frac{W}{m^2} \right) \times P_{módulo}(W) \times e^{(T_c - 25) \times coef.Pmax} \times 10^{-6}$$

*Ecuación 4: Corrección de potencia para un módulo fotovoltaico*

- P: Potencia entregada por el panel.
- G: Irradiancia incidente en el panel.
- $P_{módulo}$ : Potencia nominal del panel.
- $T_c$ : Temperatura de celda.
- coef.Pmax: Coeficiente de temperatura para potencia. Es un dato técnico del panel.

### 6.2.3. Estructura Básica de la Función

```
def selector_principal(df_consumo, ser_modulo, dataframes_list, rend, v1,v2,
                    precio_inv, periodo_explotacion, coste_montaje,
                    coste_estructura):
```

### 6.2.4. Parámetros de Entrada

- dataframes\_list: Lista de DataFrames que contienen los datos meteorológicos para diferentes orientaciones.
- rend: Rendimiento del sistema.
- precio\_inv: Precio del inversor seleccionado.
- periodo\_explotacion: Periodo de explotación del sistema en años.
- coste\_montaje: Sobrecoste debido al montaje de la instalación.
- coste\_estructura: Coste por estructura de soporte de los paneles.

### 6.2.5. Funciones y Cálculos Realizados

- Inicialización de variables: En esta sección, se inicializan varias variables necesarias para los cálculos posteriores. Estas variables incluyen los coeficientes de temperatura, potencia nominal del panel y precio del panel.
- Inicialización de listas: Aquí se inicializan listas que se utilizarán para almacenar los resultados de los cálculos para cada orientación.
- . Bucle while: Este bucle itera sobre las diferentes orientaciones almacenadas en 'dataframes\_list'.

- . Inicialización de listas y DataFrame: Se inicializan listas y un DataFrame que almacenarán los resultados de los cálculos para cada número de placas en la orientación actual.
- Cálculo de temperatura y potencia entregada: Se calcula para cada momento, la temperatura de las celdas solares y la potencia entregada por cada placa, en función de los datos meteorológicos y las características del módulo solar seleccionado.
- Bucle for: Este bucle itera sobre el rango de valores para el número de placas.
  - Copia del DataFrame: Se realiza una copia del DataFrame anterior para evitar modificar los datos originales y que surjan errores de cálculo durante la ejecución del código.
  - Cálculos adicionales: Se realizan cálculos adicionales, como la potencia entregada por la instalación, la energía aprovechada y el coste anual asociado a equipos y montaje.
  - Almacenamiento de resultados: Se almacenan los resultados en las listas correspondientes.
- . Obtención de resultados finales para la orientación actual: Se selecciona el número de placas que minimiza el precio del kWh y se almacenan los resultados en las listas correspondientes.
- Obtención de los resultados finales para cada una de las orientaciones.

#### 6.2.6. Observaciones

Es posible dimensionar la instalación en función de diferentes criterios. A continuación, se expondrán los que han sido considerados a lo largo de la creación de la sección actual.

- Minimización del importe de la factura eléctrica.
  - Ventajas: Ventajas en términos monetarios para el/los usuarios de la instalación, ya que esta opción es la que más dinero les permite ahorrar a lo largo de la vida útil de la instalación.
  - Desventajas: Riesgo de sobredimensionamiento de la instalación. Este criterio tiene en cuenta, tanto la energía tomada de la red como su precio. Por tanto, ante un eventual incremento el precio de la red puede resultar conveniente según este criterio, instalar una mayor cantidad de paneles, aunque la energía excedentaria aumente excesivamente.
- Minimización de excedentes o maximización de la energía aprovechada.
  - Ventajas: Puede considerarse como la más sostenible, dado que tiene en cuenta exclusivamente la energía que se aprovecha, minimizando los excedentes y limitando la contaminación asociada a la producción, transporte y reciclaje, de los equipos empleados.

- Desventajas: Desventajas en términos monetarios para el/los usuarios de la instalación, dado que no tiene en cuenta estos aspectos.
- Minimización del precio del kilovatio-hora
  - Esta opción se encuentra en un término medio entre las dos anteriores y es la implementada en este proyecto, dado que el precio del kWh se ha calculado como el cociente entre el coste de la instalación (€) y la energía aprovechada (kWh). A continuación, se muestra un gráfico, en el cual se ha evaluado el precio del kWh para una orientación concreta, en un rango de entre uno y diecinueve paneles. Se puede apreciar, que al comienzo la curva desciende rápidamente, dado aún hay bastante más energía que puede ser aprovechada. Sin embargo, llega un punto en el cual el incremento de la energía aprovechada es cada vez menor, mientras que los costes de la instalación siguen aumentando.

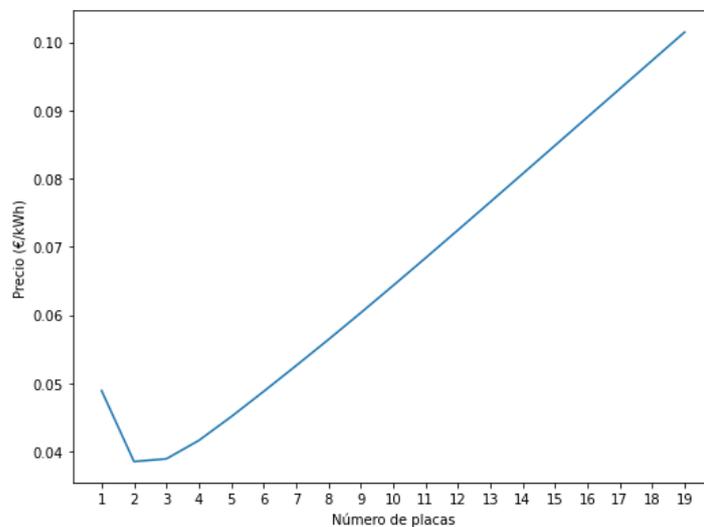


Figura 13: Precio del kilovatio-hora en función del número de paneles

### 6.3. Módulo inverter.py: Función selector\_inversor

#### 6.3.1. Propósito

El propósito de esta función es el de encontrar el inversor más adecuado para cada instalación evaluada, dentro de los modelos disponibles y en función de ciertos requisitos que se introducirán en un apartado posterior.

#### 6.3.2. Conceptos previos

##### 6.3.2.1. Función del inversor

Un inversor es un dispositivo electrónico, que se encarga de transformar la corriente continua generada por los paneles en corriente alterna, de forma que pueda ser utilizada en el hogar o vertida a la red eléctrica.

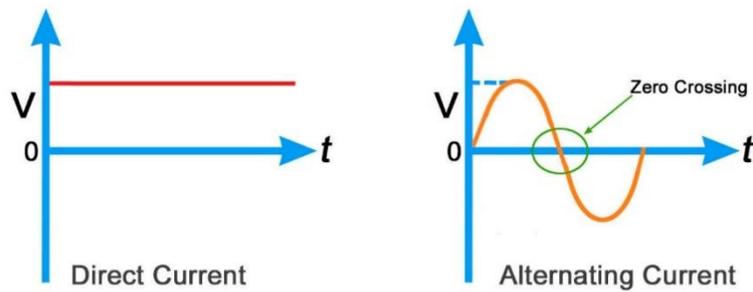


Figura 14: Corriente Continua vs Corriente Alterna. Fuente: [www.caresrenewables.com](http://www.caresrenewables.com)

La calidad de la señal puede determinarse mediante el parámetro THD (Total Harmonic Distorsion, o Distorsión Armónica Total), el cual cuantifica la cantidad de armónicos en una señal eléctrica en relación a su frecuencia fundamental.

### 6.3.2.2. MPPT (Rastreador del Punto de Máxima Potencia o Maximum Power Point Tracker)

Como se evidenció en las figuras 11 y 12 de la sección anterior, se identifica un punto de máxima potencia (PMP) correspondiente a cada combinación única de irradiancia y temperatura. El MPPT es un algoritmo sofisticado que, ajustando la tensión y empleando estrategias avanzadas de perturbación y observación, se esfuerza por localizar y mantener este PMP. El objetivo principal es optimizar la cantidad de energía que se puede aprovechar.

A continuación, se proporciona una representación gráfica del funcionamiento de un algoritmo de este tipo para mayor comprensión.

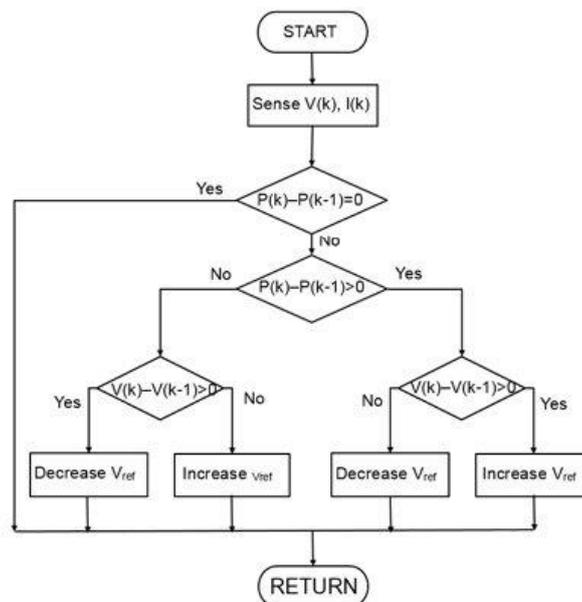


Figura 15: Esquema de Funcionamiento de un Algoritmo del Tipo Perturbar y Observar. Fuente: [www.mathworks.com](http://www.mathworks.com)

### 6.3.2.3 Rendimiento

En los documentos técnicos de los inversores es posible encontrar dos tipos de rendimientos: el rendimiento máximo y el rendimiento europeo. La diferencia entre uno y otro radica en que el

primero hace referencia al máximo rendimiento que podemos esperar del dispositivo, tal y como su propio nombre indica; el segundo, al rendimiento ponderado bajo distintos regímenes de carga y es, por tanto, más realista. En el siguiente gráfico se pueden apreciar algunas de las curvas de rendimiento de un inversor fotovoltaico.

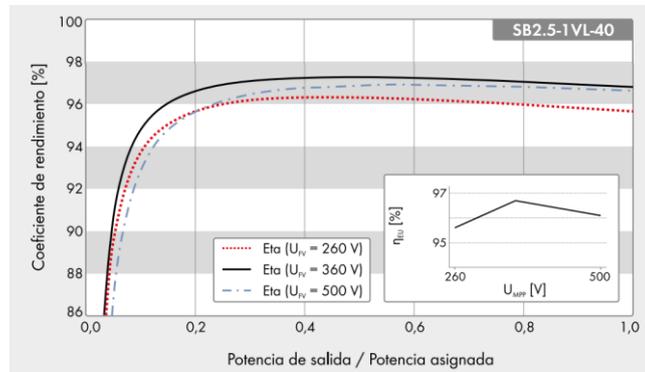


Figura 16: Coeficiente de Rendimiento frente al Cociente entre Potencia de Salida y Potencia Asignada. Fuente: Ficha Técnica del Inversor Sunny Boy 2.5 KW de SMA

Tal y como es posible observar, existen 2 factores que afectan a la eficiencia del aparato. Estos son la potencia y la tensión. Por tanto, cuanto más cerca se esté operando de las condiciones de diseño o nominales, más se acercará el rendimiento al máximo en un punto dado.

#### 6.3.2.4. Configuraciones de conexión de inversores

Existen diversas formas de conexión de inversores en sistemas fotovoltaicos. Las formas de conexión más comunes son:

- Conexión centralizada: Todos los paneles de la instalación se conectan a un mismo inversor.
  - o Ventajas: Menor coste inicial debido a la utilización de un solo inversor y menor complejidad en la instalación.
  - o Desventajas: La pérdida de rendimiento de un panel afecta a todo el sistema. Menor flexibilidad en la expansión del sistema en el futuro.
- Conexión por String: Cada una de las ramas del grupo se conecta a un inversor.
  - o Ventajas: Mayor eficiencia en sistemas con paneles solares con diferentes inclinaciones, sombreado o características.
  - o Desventajas: Mayor coste inicial y mayor complejidad en la instalación.
- Microinversores: Cada panel lleva asociado su propio inversor.
  - o Ventajas: Mayor eficiencia que la Conexión por String, en sistemas con paneles solares con diferentes inclinaciones, sombreado o características.
  - o Desventajas: Es la opción con el mayor coste inicial.

Actualmente, el programa contempla exclusivamente la forma de conexión mediante inversor centralizado.

### 6.3.3. Estructura básica de la función

```
def selector_inversor(N, df_reslt_PV_sel, df_meteo, ser_modulo, df_inv, rend):  
  
    if ...  
        return pd.Series(df_valid.loc[indice]), I_sc_max_mod * Np_sel, \  
            I_mpp_max_mod * Np_sel, V_mpp_max_mod * Ns_sel, Ns_sel, \  
            Np_sel - 1, Temp_max_seg  
  
    else:  
        return pd.Series(dtype=int), 0, 0, 0, 0, 0, 0
```

### 6.3.4. Parámetros de entrada

- N: Número de paneles fotovoltaicos.
- df\_reslt\_PV\_sel: DataFrame retornado por la función 'selector\_principal', comentada en la sección anterior.
- df\_meteo: DataFrame con los datos meteorológicos de la orientación seleccionada por la función 'selector\_principal'.
- ser\_modulo: Comentado en la sección anterior.
- df\_inv: DataFrame que contiene los datos de los inversores disponibles.
- rend: Comentado en la sección anterior.

### 6.3.5. Funciones y cálculos realizados

- Inicialización de variables
  - Temperaturas: Se extraen del DataFrame de datos meteorológicos las temperaturas, mínima y máximas registradas. A continuación, se minoran y mayoran, respectivamente, a efectos de seguridad.
  - Tensiones e intensidades: En función de las variables de temperatura inicializadas en el apartado anterior, se inicializan variables con la tensión máxima y mínima (Circuito abierto), intensidad máxima y mínima (Cortocircuito), y con las tensiones e intensidades máximas y mínimas en el punto de máxima potencia. Para ello, se utiliza la siguiente expresión.
    - $X = x \times (1 + coef * (T - 25))$
    - Donde:
      - X: Puede ser cada una de las tensiones o intensidades descritas en el apartado anterior.
      - x: Tensión de circuito abierto o en el punto de máxima potencia, o intensidad de cortocircuito o en el punto de máxima potencia, máximas o mínimas, obtenida de los datos técnicos del panel fotovoltaico.

- coef: Coeficiente de intensidad de cortocircuito o de tensión de circuito abierto, obtenida de los datos técnicos del panel fotovoltaico.
    - T: Temperatura, en función de la variable que se esté inicializando.
  - Inicialización de una tupla y un DataFrame vacío.
- Bucle for: Dado un número de placas, estas se pueden conectar siguiendo diferentes configuraciones serie-paralelo. Esta configuración, tal y como se puede verificar con las dos primeras leyes de Kirchoff, afectarán a la tensión y corriente proporcionadas por el sistema. También, por tanto, a la selección del inversor. El bucle for, en conjunto con el condicional y las dos fórmulas que se encuentran inmediatamente después, itera a través de las configuraciones posibles.
  - Inicialización de las variables de tensión y corriente de salida: Dada una configuración concreta y, por tanto, conociendo los números de paneles en serie y paralelo de la configuración, se calculan las tensiones y corrientes de salida de la instalación, relevantes para la selección del inversor.
  - Filtrado del DataFrame de inversores: Comprobación de las condiciones que han de cumplirse y filtrado del DataFrame en función de estas.
  - Cálculos adicionales: Inicialización de variables adicionales necesarias para funciones integradas en otros módulos y adición de los inversores que han pasado el filtro al DataFrame 'df\_valid'.
- Selección de un inversor: El DataFrame anterior es ordenado en función de la potencia nominal, la proximidad a la tensión nominal (Rendimiento) y el precio. Una vez ordenado, se extrae el primero, es decir, en caso de que para la potencia nominal más baja haya más de un inversor, se selecciona el que vaya a trabajar más cerca de la tensión nominal. De igual manera para el precio.

## 6.4. Módulo acs.py

### 6.4.1. Propósito

Existen varias maneras de gestionar los excedentes que se producen en una instalación fotovoltaica. Las más comunes son verterlos a la red de distribución eléctrica o almacenarlos en baterías. No obstante, es posible plantearse otras maneras de gestionarlos. El propósito de este módulo es el de simular, de manera simplificada, la utilización de excedentes para calentar el ACS de la vivienda mediante el sistema de aerotermia.

Para ello, se utilizan cuatro funciones, las cuales serán explicadas en los próximos apartados.

## 6.4.2. Funciones

### 6.4.2.1. Función *arreglo\_acs*

#### 6.4.2.1.1. Propósito

El aprovechamiento de excedentes para la producción de ACS solo se contempla dentro de un mismo día, es decir, los excedentes de un día en concreto solo podrán utilizarse para el calentamiento de la cantidad de ACS prevista para ese día.

Dado este requerimiento, esta función devuelve una lista con las fechas contenidas en el DataFrame de consumo eléctrico, para su posterior utilización.

#### 6.4.2.1.2. Estructura básica de la función y parámetros de entrada

```
def arreglo_acs(df_consumo):  
    ...  
    return df_cons, lista_fechas
```

### 6.4.2.2. Función *perdidas\_deposito*

#### 6.4.2.2.1. Propósito

Dado el objetivo del módulo actual de redistribuir la producción de ACS, es necesario contemplar las pérdidas de energía en forma de calor que sufre el depósito, debido al calentamiento del agua en un momento anterior, y su impacto en el consumo eléctrico. Para ello, es posible utilizar diferentes modelos. Esta función, proporciona un método para calcular estas pérdidas. Para ello, se ha utilizado como apoyo la “Guía técnica para diseño y cálculo del aislamiento térmico de conducciones, aparatos y equipos” del “Instituto para la Diversificación y Ahorro de la Energía”.

#### 6.4.2.2.2. Conceptos previos: Mecanismos de transferencia de calor

Transferencia de calor (o calor) es la energía en tránsito debido a una diferencia de temperaturas (Incropera y DeWitt, 1999). Existen tres formas de transferencia de calor: conducción, convección y radiación.

- **Conducción:** Es la transferencia de energía de las partículas más adyacentes de una sustancia hacia las adyacentes menos energéticas, como resultado de interacciones entre esas partículas.
- **Convección:** Es el modo de transferencia de energía entre una superficie sólida y el líquido o gas adyacentes que están en movimiento y comprende los efectos combinados de la conducción y el movimiento de fluidos.
- **Radiación:** Es la energía emitida por la materia en forma de ondas electromagnéticas (o fotones) como resultado de los cambios en las configuraciones electrónicas de los átomos o moléculas. A diferencia de las dos formas anteriores, no requiere de la presencia de un medio interventor (Çengel, 2007).

#### 6.4.2.2.3. Fórmulas empleadas

Teniendo en cuenta los modos de transferencia de calor explicados anteriormente y mediante ecuaciones simplificadas, pero aplicables a este ámbito, de la Teoría del Calor, se procederá como explicado en la introducción, a modelar la interacción del acumulador de ACS con el medio que le rodea.

#### 6.4.2.2.3.1. Resistencia térmica por conducción de una capa cilíndrica

$$R_{cond_{cil}} = \frac{\ln\left(\frac{r_{ext}}{r_{int}}\right)}{2\pi k}$$

*Ecuación 5: Resistencia térmica por conducción de una capa cilíndrica*

- $r_{int}$ : Radio interior. Distancia entre el centro del depósito y el comienzo de la capa de aislante, medida de forma paralela al plano contenido en el extremo inferior o superior del depósito.
- $r_{ex}$ : Radio exterior. Distancia entre el comienzo y el final de la capa de aislante. Se puede calcular como la suma del radio interior y el espesor de la capa de aislante.
- $k$ : Conductividad térmica de la capa de aislante.

La resistencia térmica de un material representa la capacidad de este, a oponerse al flujo de energía en forma de calor a su través. La expresión anterior es una versión particularizada para una capa cilíndrica.

#### 6.4.2.2.3.2. Coeficiente de convección exterior con aire

$$h_{conv} = 1.32 \sqrt[4]{\frac{\Delta T}{H}}$$

*Ecuación 6: Coeficiente de convección exterior para conductos cilíndricos dispuestos de forma vertical, para régimen laminar ( $D^3\Delta T \leq 10 \text{ m}^3\text{°C}$ )*

- $\Delta T$ : Diferencia de temperaturas entre la pared y el aire.
- $H$ : Anchura de la superficie, medida en metros.

Determinado experimentalmente, el coeficiente de calor cuantifica la transferencia de calor entre una superficie sólida y un fluido circundante mediante convección. Depende de la estructura física del sistema, del tipo de flujo y de la diferencia de temperaturas, entre otros.

#### 6.4.2.2.3.3. Coeficiente de radiación

$$h_{rad} = \varepsilon\sigma(T_{sup} + T_{aire})(T_{sup}^2 + T_{aire}^2)$$

*Ecuación 7: Coeficiente de convección equivalente en radiación simplificado*

- $\varepsilon$ : Emisividad de la superficie.
- $\sigma$ : Constante de Boltzmann ( $5.67 \times 10^{-8}$ ).

Esta ecuación, proveniente de la “Ley de Stefan-Boltzmann”, una expresión simplificada utilizada para calcular el flujo de calor mediante un coeficiente de convección equivalente en radiación. Se basa en la suposición de que las temperaturas superficiales del resto de superficies son aproximadamente iguales a la del aire. Esta aproximación permite simplificar el cálculo del flujo de calor por radiación en aplicaciones prácticas, donde no se tiene conocimiento de la temperatura de dichas superficies. La ecuación se deriva linealizando la relación entre las temperaturas y considerando la diferencia de temperaturas entre la pared y el aire.

#### 6.4.2.2.3.4. Resistencia térmica por convección de una capa cilíndrica

$$R_{conv} = \frac{1}{2\pi r_{ext} h_{conv}}$$

*Ecuación 8: Resistencia térmica por convección de una capa cilíndrica*

- $h_{conv}$ : Coeficiente de convección. Dado que se ha definido un coeficiente de convección equivalente en radiación, este coeficiente vendrá dado por el anterior, sumado al coeficiente de convección exterior con aire.

#### 6.4.2.2.3.5. Ecuación de flujo de calor en régimen estacionario

$$\frac{q}{H} = \frac{\Delta T}{R_{total}}$$

*Ecuación 9: Estimación del flujo de calor normalizado por altura*

- $\frac{q}{H}$ : Flujo de calor por metro.
- $\Delta T$ : Diferencia de temperaturas entre el agua y la temperatura ambiente.
- $R_{total}$ : Suma de las resistencias térmicas conductiva y convectiva.

La ecuación anterior es una forma simplificada de la “Ley de Fourier” aplicada a un sistema de transferencia de calor en régimen estacionario. Para el caso que nos ocupa, representa la pérdida de calor por metro del depósito de ACS.

#### 6.4.2.2.4. Estructura básica de la función

```
def perdidas_deposito(r_int, espesor, k, em, L, T_agua, tramo, T_amb):
...
return (q * 1e-3 * tramo)
```

#### 6.4.2.2.5. Parámetros de entrada

- $r_{int}$ : Radio interior del depósito (m).
- $espesor$ : Espesor de la capa de aislante (m).

- k: Conductividad (W/mK).
- em: Emisividad del material de la superficie exterior del depósito.
- L: Longitud del depósito (m).
- T\_agua: Temperatura del agua del depósito (K).
- tramo: Intervalos de tiempo para los cuales se calcula la energía perdida en forma de calor. Para este caso particular toma un valor de 0.5, dado que el DataFrame contiene valores promedio para cada media hora.
- T\_amb: Temperatura del aire (K).

#### 6.4.2.2.6. Funciones y cálculos realizados

- Inicialización de variables: Se inicializan las variables 'T\_sup'; necesaria para comenzar el proceso iterativo y 'cte\_boltz', con el valor aproximado de la constante de Boltzmann.
- Cálculo de la resistencia térmica conductiva para una capa cilíndrica
- Inicialización de los parámetros para la iteración: Tolerancia y número máximo de iteraciones.
- Bucle for: Proceso iterativo
  - o Cálculos adicionales: Aplicación de las fórmulas explicadas en el apartado "Fórmulas empleadas", cálculo de flujo de calor y actualización de la temperatura de la superficie.

#### 6.4.2.2.7. Observaciones

Es importante comentar, que el proceso utiliza formulas simplificadas debido a la complejidad y coste computacional de utilizar métodos numéricos.

Por otra parte, considera despreciables el coeficiente de convección interior; y por tanto la resistencia térmica asociada, y la resistencia térmica del material del depósito, debido a la alta conductividad de estos.

Por último, siendo esta la mayor de las concesiones, el proceso calcula las pérdidas estáticas del depósito. Si bien las demás aproximaciones utilizadas son suficientemente precisas, esta última puede llevar a mayores diferencias con el resultado real. Esto es debido a que, en un caso real, la temperatura del agua en el interior del depósito disminuye tanto por pérdidas de calor como por el mezclado con agua a menor temperatura, durante el proceso de vaciado-llenado, debido al consumo de ACS.

Para el caso de la disminución de temperatura debido a pérdidas de calor, es posible imaginarse que el sistema de aerotermia es capaz de activarse y contrarrestar los efectos de las pérdidas de calor lo suficientemente rápido, para considerar la temperatura del agua constante.

Sin embargo, para el caso de del mezclado con agua proveniente de la red, se comete una concesión, seguramente demasiado grande. Al no considerar la disminución de temperaturas en el depósito debido a este factor, la pérdida obtenida será mayor que la real. Es posible, de cara al futuro, ampliar el código para considerar este factor o factores. Para ello, será necesario introducir algunas

ecuaciones más y trabajar con una base de datos con mediciones en intervalos bastante más cortos de tiempo.

Finalmente, se muestran dos gráficas con los resultados obtenidos al analizar la influencia de la variación de la temperatura, así como del espesor o conductividad del aislante sobre las pérdidas.

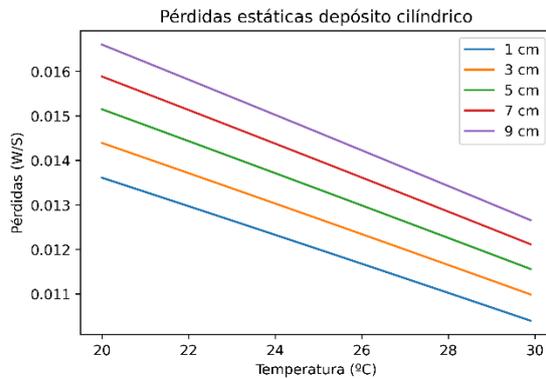


Figura 17: Pérdidas en función del espesor

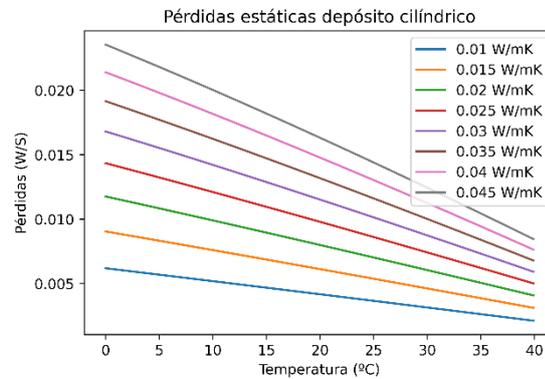


Figura 18: Pérdidas en función de la conductividad

### 6.4.2.3. Función strat

#### 6.4.2.3.1. Propósito

Esta función ha sido definida en este momento, a fin de modularizar el código y facilitar su comprensión y reutilización. Su objetivo, es el de tener en cuenta el impacto de las pérdidas sobre los sistemas de calefacción y refrigeración.

#### 6.4.2.3.2. Consideraciones iniciales

Tal y como se ha explicado anteriormente, la función 'perdidas\_deposito' devuelve el consumo eléctrico asociado a las pérdidas de calor en el depósito. Aunque difiera ligeramente de la realidad, para la simulación en DesignBuilder se ha tomado un SCOP idéntico para la producción de ACS y para el sistema de calefacción.

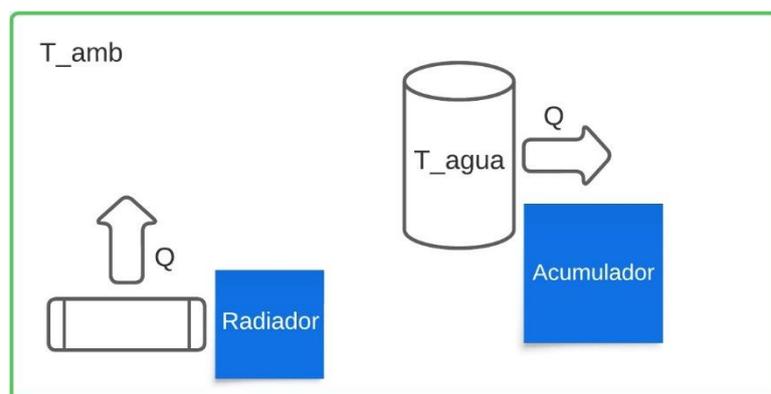


Figura 19: Interacción entre el radiador y el depósito de ACS dentro de la envolvente térmica de una vivienda

Tomando la envolvente térmica de la vivienda como la superficie de contorno y dada la consideración anterior: Durante la época de uso de calefacción (invierno), una pérdida de calor en

el acumulador de ACS supone una ganancia de calor para la vivienda, que el sistema de calefacción ya no tendrá que aportar. Por el contrario, durante la época de uso de refrigeración (verano), por el contrario, aparte de tener que aportar energía para mantener el agua del depósito a la temperatura de consigna, es necesario contrarrestar el incremento de temperatura debido a la pérdida de calor en el depósito, con el sistema de refrigeración.

#### 6.4.2.3.3. Estructura básica de la función

```
def strat(df_cons_p, lista_reha_completa):  
    lista_reha_series = pd.Series(lista_reha_completa, index=df_cons_p.index)  
    ...  
    return df_cons_p
```

#### 6.4.2.4. Función 'redistribuidor\_acs'

##### 6.4.2.4.1. Propósito

Es la función principal del módulo, haciendo uso de las dos funciones anteriores y código adicional, se encarga de reasignar la producción de ACS a los periodos de excedente.

##### 6.4.2.4.2. Estructura básica de la función

```
def redistribuidor_acs(df_cons, lista_fechas, df_rslt_PV_sel, args_fijos,  
                      df_estancia_acum):  
    ...  
    return df_cons, df_cons_p, np.array(lista_reha_completa) * args_fijos[-1]
```

##### 6.4.2.4.3. Parámetros de entrada

- df\_cons: DataFrame con los datos de consumo eléctrico, previamente procesado por "arreglo\_acs".
- lista\_fechas: Lista con las fechas contenidas en el DataFrame de consumo eléctrico, obtenida de la función 'arreglo\_acs'.
- df\_rslt\_PV\_sel: DataFrame con los resultados de los cálculos, de la configuración del sistema fotovoltaico elegida, obtenida de la función 'selector\_principal'
- args\_fijos: Lista con los parámetros necesarios para la función "perdidas\_deposito"
- df\_estancia\_acum: DataFrame con datos de temperaturas de la estancia en la que se encuentra el acumulador de ACS.

##### 6.4.2.4.4. Funciones y cálculos realizados

- Creación de una columna en el DataFrame, con los momentos de déficit (-) y excedente (+).
- Inicialización de dos listas vacías.
- Bucle for: Itera sobre cada una de las fichas contenidas en 'lista\_fechas'

- Inicialización de una matriz nula 48x48: Esta, contendrá datos de la cantidad de energía eléctrica, que se ha desplazado de un momento a otro, debida a la redistribución.
  - Inicialización de otras variables: Entre ellas dos DataFrame; con los datos de consumo y temperatura, filtrados para la fecha sobre la cual se está iterando, varios vectores; con los datos de excedente y déficit, entre otros; e índices.
  - Inicialización de una variable limitadora: Esta, permite acotar número de intervalos que se permite desplazar la producción de ACS
  - Bucle while: Dentro del cuerpo de este bucle, se encuentra la lógica mediante la cual se redistribuye el consumo de ACS. Se ejecuta mientras queden excedentes que puedan ser reasignados, es decir, mientras el periodo de excedente sea anterior al periodo de consumo.
  - Cálculo de pérdidas por redistribución: Haciendo uso de la función “perdidas\_deposito” y de la matriz, se calculan las pérdidas eléctricas debidas a la producción de ACS en un momento anterior y a su almacenamiento.
  - Cálculo y adición de los resultados, a las dos listas creadas anteriormente. Estas, se van completando con cada iteración del bucle for.
- Adición de una nueva columna al DataFrame de consumos: Necesaria para la función “strat”.
  - Modificación de las columnas de consumo: Empleo de la función “strat”
  - Recálculo de la columna ‘Total (Electricidad)’: Debida a las modificaciones hechas por la función “strat” en las columnas ‘Calefacción (Electricidad)’, ‘Refrigeración (Electricidad) y ‘ACS (Electricidad)’.

### 6.4.3. Gráficos

En primer lugar, se pueden observar las curvas de excedentes para cuatro fechas significativas a lo largo del año. Estos días representan los equinoccios de primavera y otoño, y los solsticios de verano e invierno.

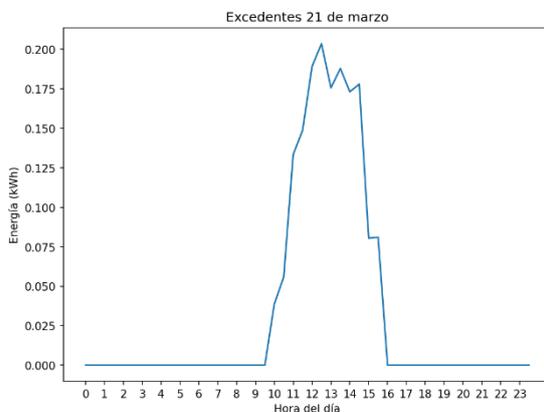


Figura 20: Excedentes del 21 de Marzo

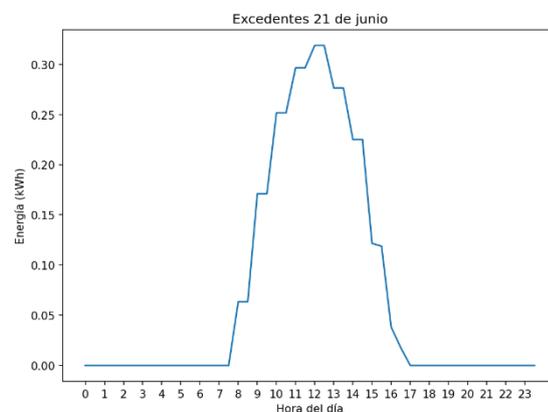


Figura 21: Excedentes del 21 de Junio

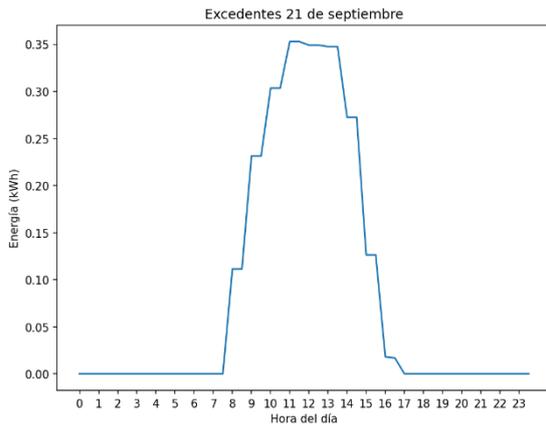


Figura 22: Excedentes del 21 de Septiembre

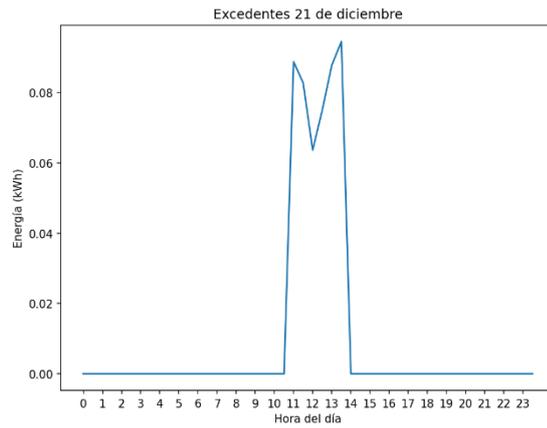


Figura 23: Excedentes del 21 de Diciembre

En el siguiente gráfico, puede apreciarse la energía excedentaria, para cada uno de los meses del año. Esta, es máxima para el mes de Junio, y decrece conforme se avanza hacia los extremos. Esto tiene que ver con que la producción de energía es máxima en los meses de verano. No obstante, pueden darse casos, en los que el mayor excedente se produzca en otros meses del año, dado que este depende tanto de la generación como del consumo.

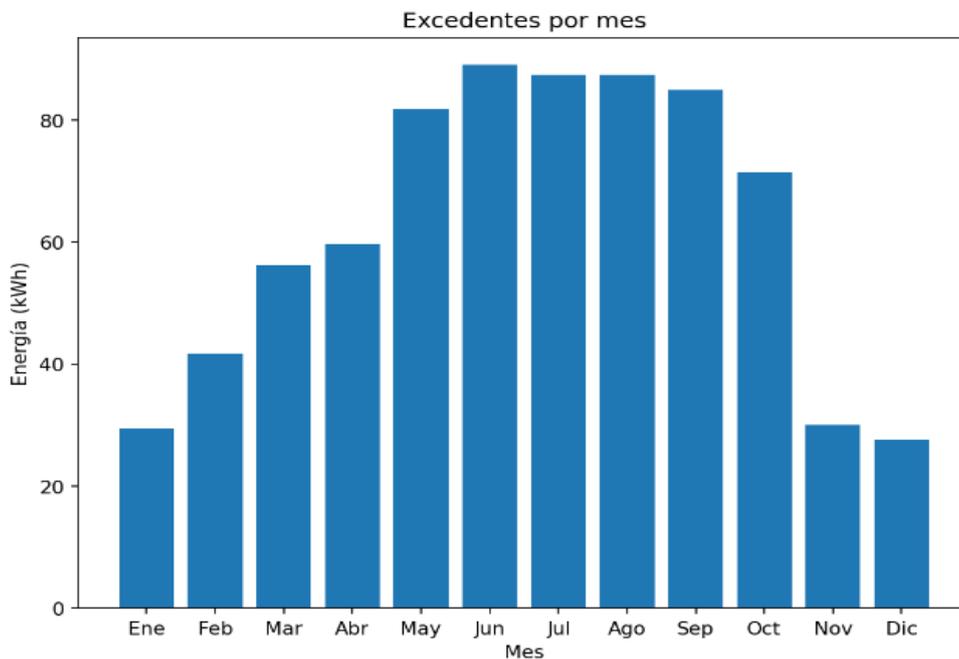


Figura 24: Excedentes por Mes

A continuación, se muestra la curva de consumo de ACS para un día promedio.

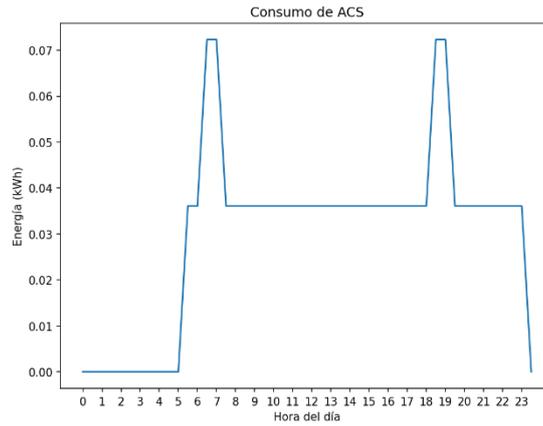


Figura 25: Curva de Consumo de ACS de la Vivienda

Tras la ejecución del proceso, y antes de añadir el consumo adicional debido al desplazamiento temporal de la producción de ACS, las curvas de producción adquieren la siguiente forma.

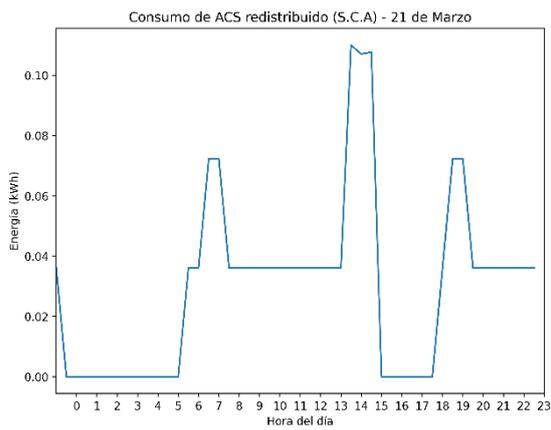


Figura 26: Curva de Consumo tras Ejecución (21M) (S.P)

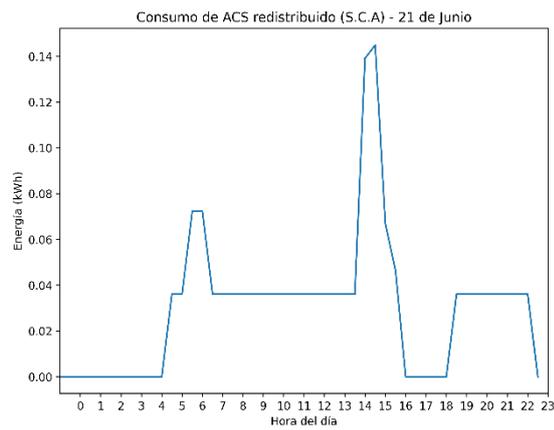


Figura 27: Curva de Consumo tras Ejecución (21J) (S.P)

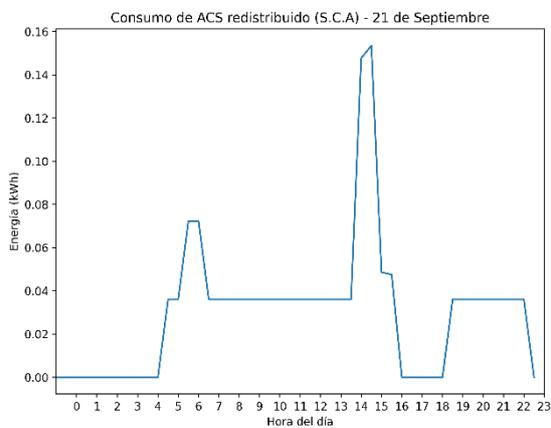


Figura 28: Curva de Consumo tras Ejecución (21S) (S.P)

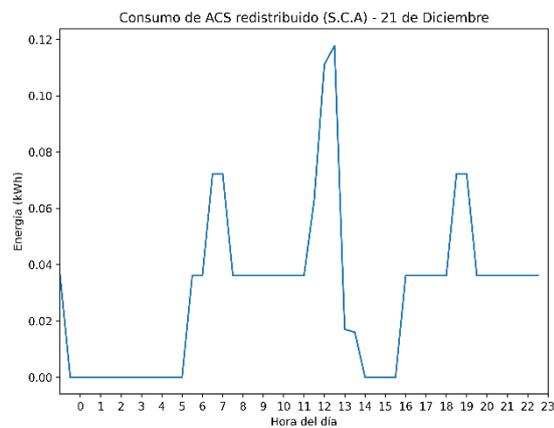


Figura 29: Curva de Consumo tras Ejecución (21D) (S.P)

Una vez añadidas las pérdidas, el resultado es el siguiente:

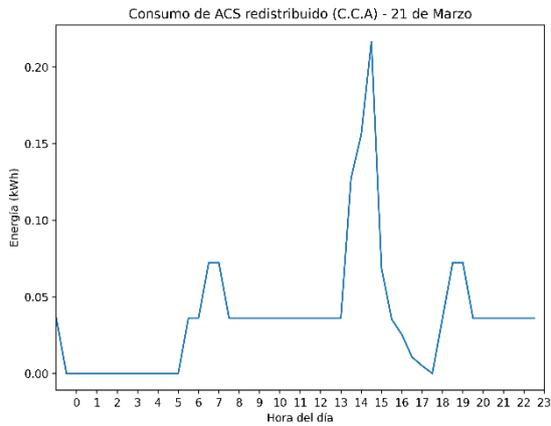


Figura 30: Curva de Consumo tras Ejecución (21M) (C.P)

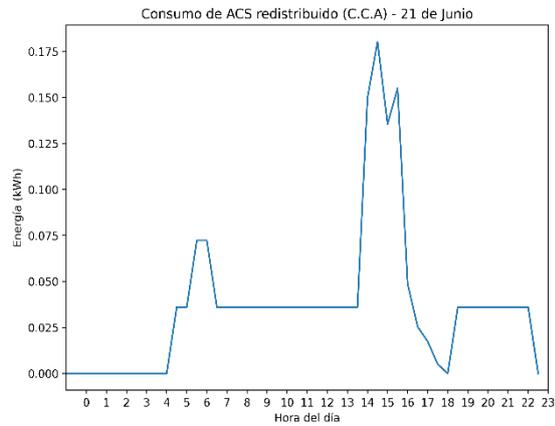


Figura 31: Curva de Consumo tras Ejecución (21J) (C.P)

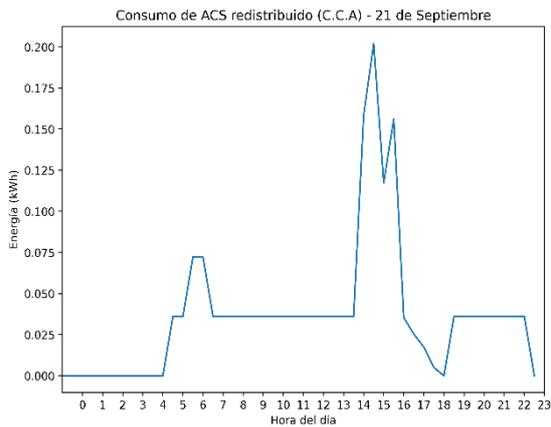


Figura 32: Curva de Consumo tras Ejecución (21M) (C.P)

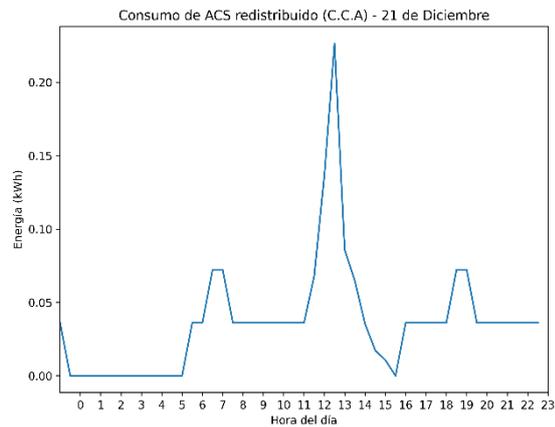


Figura 33: Curva de Consumo tras Ejecución (21J) (C.P)

Finalmente, se presenta un gráfico de barras que ilustra el consumo eléctrico adicional acumulado cada mes. Cabe destacar que este gráfico fue generado antes de ejecutar la función 'strat'. Por lo tanto, no se han incorporado las correcciones resultantes de la influencia de 'strat' en los sistemas de calefacción y refrigeración.

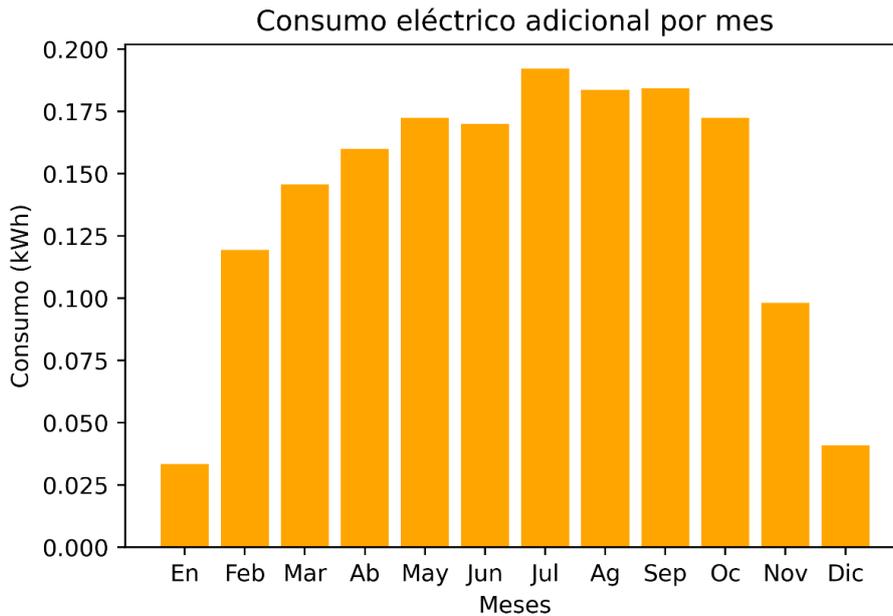


Figura 34: Consumo Eléctrico Adicional por Mes

Si bien en los meses de invierno la diferencia de temperatura entre el agua del depósito y la estancia en la que se encuentra es algo mayor, el excedente disponible es mucho menor y, por tanto, también la cantidad de energía asociada a la producción de ACS que se puede desplazar.

#### 6.4.4. Observaciones

La implementación de este módulo ha sido diseñada principalmente con un enfoque analítico, en lugar de centrarse en la instauración de un sistema de redistribución de ACS. Si bien es innegable que parte de los excedentes de la instalación se aprovechan, también se incrementan el consumo eléctrico y las pérdidas en el depósito. Esto podría traducirse en un beneficio económico, como se discutirá en apartados posteriores. Por tanto, es pertinente explorar otras alternativas para el empleo de estos excedentes como pueden ser la inyección en la red eléctrica o la automatización de tareas, como puede ser la activación programada de electrodomésticos, y establecer un orden de prioridades en función del beneficio económico y los pros y contras asociados a cada una de las alternativas. Mediante la combinación de diversas técnicas, es posible encaminarse hacia un futuro en el que la energía se administre de una forma cada vez más eficiente.

Es relevante destacar que este estudio se centra únicamente en la redistribución de la producción de ACS. Sin embargo, debido al consumo energético significativo asociado, resulta sumamente intrigante investigar la viabilidad de administrar tanto la demanda como la producción de los sistemas de climatización.

## 6.5. Módulo electrico.py: Función calcular\_S

### 6.5.1. Propósito

En estos momentos, este módulo integra una única función, la cual se encarga de calcular la sección de los conductores. Es posible implementar más adelante, otras funciones que se encarguen de calcular las características de la aparamenta eléctrica de protección.

Esta función, es llamada desde el módulo principal y se encarga de calcular las secciones de los conductores de los tramos de continua y alterna, haciendo uso de la base de datos de intensidades admisibles.

### 6.5.2. Conceptos previos

A continuación, se da una breve explicación de los dos criterios adoptados para el cálculo de la sección de un conductor.

#### 6.5.2.1. Criterio térmico

Es un criterio de seguridad y se centra en evitar el calentamiento excesivo de un cable debido a la corriente que circula a través de él, dado que la corriente eléctrica provoca una disipación de energía en forma de calor.

#### 6.5.2.2. Criterio de caída de tensión

Es un criterio de servicio y en instalaciones fotovoltaicas, se adopta para asegurar el rendimiento óptimo del sistema y maximizar la eficiencia de conversión de energía solar a eléctrica.

Para instalaciones situadas en España, los límites de caída de tensión son del 1,5 %, según lo dispuesto en el punto 5 de la ITC-BT-40 del Reglamento Electrotécnico de Baja Tensión y en el Pliego de Condiciones Técnicas de Instalaciones Conectadas a Red del IDAE.

### 6.5.3. Fórmulas empleadas

#### 6.5.3.1. Temperatura de servicio

$$T_{servicio} = T_{ambiente\_max} + (T_{servicio\_permanente\_max} - T_{ambiente\_max}) \times \left( \frac{I_{mpp}}{I_{max\_cable}} \right)^2$$

*Ecuación 10: Temperatura de servicio de un conductor*

- $T_{ambiente\_max}$ : Temperatura ambiental máxima
- $T_{servicio\_permanente\_max}$ : Temperatura máxima admisible en servicio permanente. Este límite está determinado por el tipo de aislante utilizado y puede variar según las características específicas del material aislante utilizado.
- $I_{mpp}$ : Intensidad generada por la instalación en el punto de máxima potencia.
- $I_{max\_cable}$ : Intensidad máxima admisible por el cable.

Mediante la expresión anterior, se calcula la temperatura alcanzada por el cable durante el funcionamiento estándar de la instalación, específicamente en el punto de máxima potencia.

#### 6.5.3.2. Caída de tensión

$$cdt = \frac{\rho_{T_{servicio}} \times L \times I_{mpp}}{S \times V_{mpp}} \times 100$$

Ecuación 11: Caída de tensión porcentual en un conductor

- $\rho_{T_{servicio}}$ : Resistividad del conductor. Es calculada de la siguiente manera:

- $\rho_{T_{servicio}} = \rho_{T=20^{\circ}C} \times (1 + \alpha_{conductor} \times (T_{servicio} - 20))$

Ecuación 12: Resistividad de un conductor a la temperatura de servicio

- $\rho_{T=20^{\circ}C}$ : Resistividad del conductor a 20°C.
    - $\alpha_{conductor}$ : Coeficiente de temperatura de la resistencia eléctrica. Describe cómo varía la resistencia de un material conductor en función de los cambios de temperatura.
  - $L$ : Longitud del cable.
  - $S$ : Sección del cable.
  - $V_{mpp}$ : Tensión alcanzada por la instalación en el punto de máxima potencia.

#### 6.5.4. Estructura básica de la función

```
def calcular_S(metodo, aislante, L, I_max, I_mpp_inst, V_mpp_inst, T_amb_max,
              f_corr=1):
    ...
    return S, I_max_cable, cdt, R
```

#### 6.5.5. Parámetros de entrada

- **metodo**: Método de instalación del cable.
- **aislante**: Material aislante del cable.
- **L**: Longitud del cable.
- **I\_max**: Intensidad máxima que circulará a través del cable.
- **I\_mpp\_inst**: Intensidad que circulará por el cable cuando la instalación funcione en el punto de máxima potencia.

- V\_mpp\_inst: Voltaje que entregará la instalación cuando funcione en el punto de máxima potencia.
- T\_amb\_max: Temperatura ambiente máxima en el lugar en el que se encuentra la instalación.
- f\_corr: Factor de corrección conjunto, sin incluir el factor de corrección por temperatura, que es calculado a lo largo del código.

#### 6.5.6. Funciones y cálculos realizados

- Inicialización de datos: Se inicializan dos DataFrame con las bases de datos que contienen las intensidades admisibles.
- Selección de datos relevantes: Se localizan en las bases de datos las fila y columna correspondientes al método de instalación y al tipo de aislante elegidos.
- Obtención del factor de corrección por temperatura: Haciendo uso de la información de la Hoja 3 de la base de datos y de la temperatura ambiente máxima, se determina el factor de corrección por temperatura.
- Amplificación de la intensidad máxima: Se incrementa la intensidad máxima de la instalación en un 25% para proporcionar un margen de seguridad, de acuerdo con el punto 5 de la ITC-BT-40 del Reglamento Electrotécnico de Baja Tensión. Aparte, se incrementa el factor de corrección por de temperatura previamente calculado, y con el factor de corrección adicional obtenido por parámetro.
- Preparación para iteraciones: Se inicializan dos variables requeridas para la implementación de los bucles "while".
- Iteración para la determinación de la sección: Se ejecutan dos bucles anidados del tipo "while". Mediante el uso de los DataFrame con los datos de intensidades admisibles y los datos proporcionados como parámetros, se aplica un conjunto de fórmulas para determinar la sección que cumple con los criterios térmicos y de caída de tensión.

#### 6.5.7. Observaciones

Es muy importante tener en cuenta, que el cálculo de la sección se ha efectuado exclusivamente, en función de los criterios térmico y de caída de tensión. Dado que toda instalación lleva asociada aparamenta eléctrica de protección, es posible que, durante el dimensionado de esta, sea necesario incrementar la sección calculada.

### 6.6. Módulo `modulos_tfg`

#### 6.6.1. Propósito

En este módulo se han reunido las funciones más básicas con el fin de organizar el programa y facilitar su comprensión.

## 6.6.2. Funciones

### 6.6.2.1. Función *procesar\_archivos*

#### 6.6.2.1.1. Propósito

Tiene dos parámetros de entrada: 'consumo\_compartido' y 'ruta'. En caso de que la primera contenga "N", lee el archivo CSV contenido en la ruta indicada y lo convierte a DataFrame. En caso de que 'consumo\_compartido' sea igual a "S", lee todos los archivos CSV contenidos en la ruta indicada, y devuelve un DataFrame con el mismo número de filas y columnas que cada uno de los archivos, pero cuyos valores son la suma de todos ellos.

#### 6.6.2.1.2. Estructura Básica de la Función

```
def procesar_archivos(consumo_compartido, ruta):  
    ...  
    if...  
    return df_consumo_fix  
    ...  
    else...  
    return df_consumo_fix
```

### 6.6.2.2. Función *coste\_instalacion\_anual*

#### 6.6.2.2.1. Propósito

Devuelve el coste anual total de la instalación, incluido el montaje.

#### 6.6.2.2.2. Estructura Básica de la Función

```
def coste_instalacion_anual(n_paneles, precio_panel, coste_estructura,  
                           precio_inversor, periodo_explotacion,  
                           coste_montaje):  
    ...  
    return coste_total_anual
```

### 6.6.2.3. Función *perdidas\_modulo*

#### 6.6.2.3.1. Propósito

Devuelve las pérdidas medias del módulo, a lo largo de la vida útil estimada, haciendo uso de los datos técnicos del panel.

#### 6.6.2.3.2. Estructura Básica de la Función

```
def perdidas_modulo(ser_modulo, periodo_explotacion):  
    ...  
    return mean(eta_anual)
```

### 6.6.2.4. Función *rend\_inst*

#### 6.6.2.4.1. Propósito

Devuelve el rendimiento de la instalación, teniendo en cuenta tanto valores usuales, como datos de los equipos utilizados y del cableado.

#### 6.6.2.4.2. Estructura Básica de la Función

```
def rend_inst(ser_modulo, ser_inv, R_DC, R_AC, df_meteo_sel, Vmpp_med,
              periodo_explotacion):
...
return rend
```

#### 6.6.2.5. Función calcular\_importe

##### 6.6.2.5.1. Propósito

Se utiliza para calcular el importe de la factura eléctrica, en función del precio de la energía de la red, de la energía de la instalación fotovoltaica y de la fracción que se toma en cada momento de cada una de ellas.

##### 6.6.2.5.2. Estructura Básica de la Función

```
def calcular_importe(fila, precio_ekWh_inst, precio_ekWh_red):
...
if...
...
elif...
...
return - fila['Dif.'] * precio_ekWh_red + (
        fila['Total (Electricidad)'] + fila['Dif.']) * precio_ekWh_inst
```

#### 6.6.2.6. Función distancia

##### 6.6.2.6.1. Propósito

Devuelve la distancia a dejar entre paneles fotovoltaicos.

##### 6.6.2.6.2. Estructura Básica de la Función

```
def distancia(ser_modulo, inclinacion, latitud):
...
return altura_modulo_inclinado * k * 1e-3
```

#### 6.6.2.7. Función pasar\_a\_K

##### 6.6.2.7.1. Propósito

Convierte grados Celsius en grados Kelvin.

##### 6.6.2.6.2. Estructura Básica de la Función

```
def pasar_a_K(columna):
...
return columna + 273
```

## 6.7. Módulo 'simulador.py'

### 6.7.1. Propósito

Este módulo incluye dos funciones mediante las cuales, es posible simular la distribución de la energía excedentaria de las viviendas que eventualmente podrían formar una comunidad

energética. Su objetivo principal, es el de poder evaluar de antemano si un grupo de viviendas es apto para formar tal comunidad. Dada la variedad en el dimensionamiento de la instalación fotovoltaica en cada vivienda, la diversidad en el tiempo de uso, así como la fluctuación en el consumo y los momentos de exceso de energía de cada casa, es necesario analizar la compatibilidad entre las viviendas. Esto permite una evaluación de la rentabilidad de la inversión venidera.

## 6.7.2. Funciones

### 6.7.2.1. Función `calcular_reparto`

#### 6.7.2.1.1. Propósito

Se utiliza para calcular el reparto de energía eléctrica entre varias viviendas a partir de los archivos CSV con sus consumos eléctricos. El objetivo es determinar la cantidad de energía que corresponde a cada vivienda y calcular el porcentaje que cada una debe pagar del coste total de la instalación.

#### 6.7.2.1.2. Estructura básica de la función

```
def calcular_reparto(ruta_carpeta, p_entregada):  
    ...  
    return df_final, asignacion, lista_num
```

#### 6.7.2.1.3. Parámetros de entrada

- `ruta_carpeta`: Es la ruta de la carpeta que contiene los CSV con los datos de consumo eléctrico de las viviendas.
- `p_entregada`: Objeto de Pandas del tipo Series con la potencia entregada por la instalación en cada momento.

#### 6.7.2.1.4. Cálculos realizados

- Preparación de los Archivos de Entrada: Se recopilan todos los archivos CSV en la carpeta especificada para su posterior procesamiento.
- Inicialización y Preparación de los Datos: Se inicializan las listas y se crea un DataFrame vacío, donde se almacenarán los datos totales. También se añade una columna al DataFrame con la potencia entregada por la instalación.
- Recopilación de Datos de Consumo de las Viviendas: Se leen archivos CSV que contienen información de consumo eléctrico de cada vivienda y estos se añaden al DataFrame. Aquí, también se realiza una suma del consumo eléctrico de todas las viviendas por fila.
- Cálculo de Consumo Proporcional: Se determina el porcentaje de energía consumida por cada casa en relación al total para cada periodo. Este cálculo se realiza solamente en los periodos en los que hay generación fotovoltaica para garantizar una asignación justa de energía.
- Determinación de Consumo Conjunto: Se crea una nueva columna que refleja el menor valor entre el consumo eléctrico conjunto y la energía generada por la instalación.

- Cálculo de Proporción de Energía Aprovechable: Se crean nuevas columnas que contienen el cociente entre el consumo eléctrico de cada casa y la energía que puede ser aprovechada de la instalación fotovoltaica, utilizando el mínimo calculado en el paso anterior.
- Asignación de Coeficientes de Energía: Se calculan los coeficientes de asignación de energía para cada vivienda y se añaden a una lista.
- Creación del DataFrame Final: Se inicia un nuevo DataFrame que incluye la potencia entregada por la instalación, el consumo eléctrico de cada vivienda y la energía asignada a cada vivienda en cada momento. Este DataFrame final es el producto de la energía generada y su respectivo coeficiente de asignación.

#### 6.7.2.2. Función *repartir\_excedentes*

##### 6.7.2.2.1. Propósito

El propósito de esta función es el de repartir la energía excedentaria de una vivienda o viviendas, entre las viviendas con déficit.

##### 6.7.2.2.2. Estructura básica de la función

```
def repartir_excedentes(row, asignacion, lista_num):
...
return reparto_con_ex, excedente_final
```

##### 6.7.2.2.3. Parámetros de entrada

- row: Parámetro que representa una fila de un DataFrame, a la cual se le aplicará posteriormente el método “apply” de Pandas.
- asignacion: Lista con los coeficientes de asignación de energía de cada vivienda, calculados y retornados por la función ‘calcular\_reparto’
- lista\_num: Lista con el número asignado a cada vivienda para identificarla.

##### 6.7.2.2.4. Cálculos realizados

Con el fin de brindar una explicación más clara del proceso llevado a cabo, se utilizará un ejemplo numérico sencillo. Se supone que en un periodo de tiempo t, una instalación fotovoltaica genera una cantidad de energía  $E=100$  unidades. Además, se considera que hay una comunidad energética compuesta por tres viviendas, las cuales requieren 80, 20 y 15 unidades de energía respectivamente. Asimismo, se asignan coeficientes de distribución de energía de 0.5, 0.4 y 0.1 a cada una de estas viviendas, en el mismo orden

- Conversión de los coeficientes de asignación pasados por parámetro en un arreglo Numpy.
- Creación de una lista de consumo obteniendo los valores de la columna “Total (Electricidad)” para cada elemento de ‘lista\_num’ en la fila ‘row’.
  - o Consumo de cada vivienda:

[80, 20, 15]

- Cálculo del reparto inicial multiplicando la energía entregada por la instalación en 'row' por los coeficientes de asignación.

$$100 \times [0.5, 0.4, 0.1] = [50, 40, 10]$$

- Cálculo del consumo por cubrir como la diferencia entre el consumo y el reparto inicial, tomando el valor máximo entre cero y esa diferencia.

$$[30, 0, 5]$$

- o Como se puede observar, la primera vivienda tiene un déficit de 30 unidades, la tercera de 5, mientras que la segunda tiene su consumo cubierto. La cantidad de energía no cubierta es, por tanto, de 35 unidades.
- Cálculo del excedente inicial como la diferencia entre el reparto inicial y el consumo, tomando el valor máximo entre cero y esa diferencia.

$$[0, 20, 0]$$

- Cálculo de la cantidad a reasignar, como el mínimo entre la suma del consumo por cubrir y la suma del excedente inicial.

$$20$$

- En caso de que haya energía para reasignar, cálculo de los nuevos coeficientes de reasignación como el cociente entre consumo por cubrir de cada vivienda y la suma total del consumo por cubrir. Aparte, cálculo de coeficientes que determinarán cuanta energía cede cada vivienda con excedentes.

$$[30, 0, 5] \div 35 = [0.857.., 0, 0.142..] \quad (\text{Coeficientes de derecho de consumo})$$

$$[0, 20, 0] \div 20 = [0, 1, 0] \quad (\text{Coeficientes de obligación de cesión})$$

- Esta forma de determinar los nuevos coeficientes implica tanto que la energía disponible se distribuirá de forma proporcional según las necesidades energéticas de cada vivienda, como que cada vivienda con excedentes cederá proporcionalmente en función del sobrante.
- Creación de nuevas variables con el resultado de la compartición. En una lista se obtiene el excedente final y en la otra la cantidad de energía que recibe cada vivienda finalmente.

[0, 0, 0]            (Excedentes tras proceso de compartición)

[67.142.., 20, 12.857..]            (Energía recibida tras proceso de compartición)

## 6.8. Módulo modulo\_principal

### 6.8.1. Propósito

El módulo principal sirve como núcleo del programa, coordinando las operaciones de importación de módulos, inicialización de variables, procesamiento de archivos de datos, y realización de cálculos. Es el eje central que dirige las acciones del programa, desde la lectura y formateo de los datos hasta la presentación de los resultados. Selecciona y ejecuta los módulos y funciones necesarios en función de los parámetros de entrada, y gestiona los procesos iterativos y condicionales, asegurando así el correcto funcionamiento de todo el sistema.

### 6.8.2. Cálculos y funciones realizadas

- Importación de módulos y funciones: En esta sección, se importan las funciones y módulos necesarios para el correcto funcionamiento del programa
- Inicialización de variables de configuración: Se inicializan tres variables de tipo 'string' para permitir al usuario especificar las opciones deseadas. Estas variables se utilizan para determinar qué módulos se utilizarán durante la simulación. Es importante destacar que el módulo de ACS no es compatible con el módulo de dimensionado conjunto.
- Inicialización de variables de configuración adicionales: En esta sección, se inicializan varias variables que se utilizarán en todo el programa. Estas variables deben ser configuradas por el usuario según los requisitos específicos de la instalación.
- Procesamiento de archivos de consumo: Se especifica la ruta de la carpeta que contiene los archivos CSV con los datos de consumo de las viviendas. A continuación, se utiliza la función 'procesar\_archivos' para procesar estos archivos y se redondea el número de decimales del DataFrame resultante a cuatro. Esto se realiza para evitar errores numéricos insignificantes en el consumo. Es importante destacar que la carpeta de consumos debe contener solo un archivo CSV si no se utiliza el modo 'dimensionado\_conjunto', de lo contrario, se producirá un error.

- Lectura de datos meteorológicos: En esta sección se realiza la lectura de los archivos CSV que contienen los datos meteorológicos para diferentes orientaciones. Estos archivos se encuentran en la carpeta especificada y se almacenan en una lista de DataFrames.
- Lectura de datos de paneles fotovoltaicos: Se importa el archivo Excel que contiene los datos técnicos de los paneles fotovoltaicos y se convierte en un DataFrame. A continuación, se realiza una agrupación por número de ficha técnica para evitar cálculos redundantes, ya que los paneles con la misma ficha técnica tienen características similares. Se extraen los datos del primer panel de cada grupo y se almacenan en un DataFrame separado.
- La extracción del primer panel es un criterio aleatorio y puede ser modificado, de forma que se adapte a las necesidades del usuario.
- Lectura de datos de inversores: Se lee el archivo Excel que contiene los datos técnicos sobre los inversores y se convierte en un DataFrame. Estos datos se utilizarán para seleccionar el inversor adecuado durante el proceso de dimensionamiento.
- Inicialización de variables de resultados: Se inicializan varias listas que se utilizarán para almacenar los resultados de las iteraciones del programa.
- Iteración sobre los paneles fotovoltaicos: En esta sección se inicia un bucle que recorre los módulos fotovoltaicos de la base de datos.
  - Inicialización de variables para el proceso iterativo: Se establecen las variables necesarias para el proceso iterativo posterior. El rendimiento inicial se establece en 0.86, y este valor se actualizará en cada iteración para reflejar el rendimiento real de la instalación.
  - Bucle while
    - Llamado a la función 'selector\_principal'
    - Llamado a la función 'selector\_inversor'
    - Verificación de la existencia de un inversor compatible: Se verifica si se ha encontrado un inversor que cumple con los requisitos de la configuración. En caso negativo, añade la variable "None" a las listas de resultados, se sale del bucle while y se pasa a la siguiente iteración. En caso afirmativo, se actualiza el precio del inversor de cara a obtener una estimación más precisa del precio del kilovatio-hora y se obtiene la corriente máxima de salida del inversor, que será utilizada en el cálculo de la sección del cable de corriente alterna.
    - Primer llamado a la función 'calcular\_S': De cara a obtener los resultados para el tramo de continua.
    - Segundo llamado a la función 'calcular\_S': De cara a obtener los resultados para el tramo de alterna.
    - Cálculo del rendimiento: Actualización del rendimiento inicial o de la iteración anterior, mediante el llamado a la función 'rend\_inst'. Redondeado posterior a tres decimales.

- Control de condiciones: Las operaciones del bucle cesan si se cumplen dos condiciones específicas:
  1. La diferencia absoluta entre el precio calculado en la iteración actual y el precio de la iteración anterior se encuentra por debajo de un valor de tolerancia especificado.
  2. El número de módulos calculado en la iteración actual es igual al número de módulos de la iteración anterior.

Estos controles son esenciales para garantizar que el algoritmo converge a una solución adecuada. En particular, estos evitan el fenómeno de divergencia oscilatoria que podría ocurrir en ciertos casos raros, en los que el número de módulos a instalar varía entre iteraciones consecutivas.

Como se ilustra en la Figura 13 del presente documento, dentro de un rango específico de paneles, el precio por kilovatio-hora puede ser muy similar. Esto, combinado con el hecho de que el número de paneles a instalar influye tanto en la selección del inversor como en su rendimiento, y que el precio del inversor afecta tanto el coste total de la instalación como el del kilovatio-hora, puede provocar esta oscilación.

Condicionales: El programa sale del bucle una vez que se dan dos condiciones: la diferencia entre el precio calculado durante la iteración en curso y el de la iteración anterior, se sitúa por debajo de la tolerancia; y el número de módulos obtenidos durante la iteración en curso coincide con el obtenido durante la iteración anterior.

Para mitigar este comportamiento, se introduce una variable adicional 'contador' y una variable booleana 'condicion'. En conjunto, cuando el número de iteraciones ha llegado a diez, permiten salir del bucle, utilizando la solución correspondiente al kilovatio-hora mínimo.

- Actualización de variables
  - Almacenamiento de resultados: Los resultados de la iteración actual se almacenan en las listas correspondientes para su posterior análisis y presentación.
- Extracción y almacenamiento de la configuración elegida: En este momento, una vez finalizado el bucle for, se dispone de las listas de resultados. Estas pueden albergar resultados numéricos en caso de haberse encontrado un inversor o de una variable del tipo "None". En esta sección, se extrae el índice del panel que ha minimizado el precio del kilovatio-hora, y se almacenan los resultados concernientes a este en una lista de resultados finales.
- Obtención del importe anual de la factura eléctrica: Se obtiene el importe anual de la factura eléctrica mediante el llamado a la función 'calcular\_importe'. A continuación, se agrega este a la lista de resultados finales.
- Cálculo de la distancia a dejar entre paneles: Se realiza un llamado a la función 'distancia', un redondeo a dos decimales y el agregado a la lista de resultados.
- Sección de cálculo del modo ACS

- Comprobación de las condiciones: Tal y como se ha comentado anteriormente, los modos de ACS y dimensionado conjunto no están ideados para ser ejecutados en la misma simulación. Mediante los condicionales “if-elif”, el programa se asegura de lanzar una excepción e interrumpirse, si esta condición no ha sido cumplida.
  - Preparación y formateo de datos de temperatura: Se lee desde archivo la base de datos con las temperaturas de la estancia del acumulador, se convierte a DataFrame y se procesa asegurar un formato horario adecuado y para estandarizar las unidades de temperatura.
  - Arreglo de los datos de entrada: Se construye una lista con los argumentos que serán pasados por parámetro, se llama a la función ‘arreglo\_acs’ para procesar las bases de datos de consumos eléctricos y temperatura
  - Llamado a la función ‘redistribuidor\_acs’
  - Cálculos adicionales: Se preparan los datos para ser pasados a la función ‘calcular\_importe’, se calcula el precio del kilovatio-hora y se hace el llamado a la función mencionada.
- Impresión de resultados
  - Sección de cálculo del modo dimensionado conjunto: A lo largo de toda esta sección se intercalan comandos de impresión, de forma que se vayan imprimiendo los resultados por pantalla. No se hace mención en estos comandos a lo largo del resumen siguiente.
    - Llamado a la función ‘calcular\_reparto’
    - Agrupación de los datos: Se inicializa un nuevo DataFrame en el cual se irán agrupar los datos de potencia asignada y consumo eléctrico de la vivienda que está siendo analizada.
    - Cálculo de la energía aprovechada y del precio del kilovatio-hora.
    - Preparación de los datos de entrada: Adición de una nueva columna al DataFrame, de forma que pueda ser pasado a la función ‘calcular\_importe’
    - Llamado a la función ‘calcular\_importe’
  - Sección de cálculo del modo simulación
    - Inicialización de listas: Inicialización de tres listas con los nombres de las columnas relevantes para este módulo, de cada una de las viviendas.
    - Llamado a la función ‘repartir\_excedentes’
    - Cálculo de la energía aprovechada por cada vivienda
    - Cálculos adicionales: Al igual que en apartados anteriores, se preparan los datos para ser pasados a la función ‘calcular\_importe’, se calcula el precio del kilovatio-hora y se imprimen los resultados por pantalla.

#### 6.8.4. Resultados

En esta sección, se presentan los resultados derivados de las diferentes ejecuciones del programa. Los valores de entrada para las diversas funciones se pueden consultar en la Sección 8 del Anexo 6:

Código en Python. En caso de haber realizado alguna modificación para fines de análisis, se hará una mención correspondiente a lo largo de la sección.

En esta primera imagen, pueden observarse los resultados del dimensionado de la instalación fotovoltaica para la vivienda previamente modelada mediante DesignBuilder.

```
----- RESULTADOS -----  
  
Orientación elegida: 2  
Nº ficha panel fotovoltaico: 8  
Marca del panel fotovoltaico: Ja Solar  
Modelo del panel fotovoltaico: JAM54S30  
Potencia del panel fotovoltaico: 390  
Nº de módulos fotovoltaicos a instalar: 2  
Nº de ramas en Serie: 2  
Nº de ramas en paralelo: 0  
Distancia a dejar entre modulos: 1.45 m.  
Nº ficha inversor fotovoltaico: 9  
Marca inversor fotovoltaico: Growatt  
Modelo inversor fotovoltaico: MIC TL-X  
Potencia inversor fotovoltaico: 0.75 kW.  
Sección DC: 6.0 mm2  
Caída de tensión en el tramo DC: 1.39 %  
Sección AC: 1.5 mm2  
Caída de tensión AC: 0.41 %  
Coste de la instalación: 815.16 €  
Energía entregada: 2376 kWh/año  
Energía aprovechada: 1684 kWh/año  
% Energía aprovechada: 70.87 %  
Rendimiento de la instalación: 77.5 %  
Precio de la energía de la instalación: 0.019362 €/kWh  
Importe de la factura eléctrica: 783.98 €/año
```

Figura 35: Resultados – Dimensionado instalación fotovoltaica

La ejecución del modo ACS, arroja por su parte los siguientes resultados:

Los datos de entrada de la función de la función 'perdidas\_deposito', son los incluidos en el código del Anexo 6 y pueden consultarse en la Sección 8 de este.

```
----- RESULTADOS MODO ACS -----  
  
Energía aprovechada: 1776 kWh/año  
% Energía aprovechada: 74.75 %  
Incremento de la energía eléctrica consumida 0.0396 %  
Precio de la energía de la instalación: 0.018358 €/kWh  
Importe de la factura eléctrica: 782.32 €/año
```

Figura 36: Resultados – Modo ACS

Como se ha mencionado en capítulos previos, las funciones implementadas poseen ciertas limitaciones. De igual forma, en el proceso de redistribución del ACS, se emplea una algoritmia

básica. Esta no se encuentra optimizada para minimizar el costo de la factura eléctrica, sino que simplemente reubica la producción del ACS a los momentos de excedente más cercanos.

Dados los resultados obtenidos, la estrategia no parece ser de utilidad para el caso analizado. De cara a comprender el motivo, conviene profundizar en el análisis. Para ello, se presenta en primer lugar una tabla con datos recopilados para diferentes valores de pérdidas estáticas, variando a su vez el parámetro 'distancia\_t\_max'. Para ello, de cara a disponer de algo más de energía excedentaria, se ha fijado el parámetro 'v1' en cuatro, de forma que sea cuatro el número mínimo de paneles a instalar. El número de paneles a instalar, como esperado, ha sido de cuatro, cada uno con una potencia de 390 W, haciendo un total de 1.560. Para este caso, el porcentaje de energía aprovechada ha resultado en un 43,33 %.

'distancia_t_max'	<b>Resultados</b>				
	<b>4</b>	<b>8</b>	<b>12</b>	<b>16</b>	<b>20</b>
<b>Δ energía aprovechada (%)</b>	0,71	1,66	2,57	3,36	3,67
<b>Δ energía consumida (%)</b>	0,0073	0,023	0,0401	0,0623	0,0656
<b>Δ precio (€ / MWh)</b>	-0,472	-1,09	-1,658	-2,134	-0,00232
<b>Δ importe factura (€)</b>	-0,950	-2,200	-3,34	-4,29	-4,670

Tabla 15: Impacto de la variación del parámetro de retraso temporal sobre la eficiencia energética y costos asociados

Los resultados mostrados en las tablas anteriores son congruentes. Es posible apreciar, un incremento en la energía aprovechada (y, por tanto, un decremento del precio del kilovatio-hora), por el incremento del parámetro 'distancia\_t\_max'. Lo mismo sucede con el incremento de la energía consumida. Por otra parte, el importe de la factura disminuye conforme se aumenta el parámetro de retraso temporal, no obstante, lo hace cada vez de una forma más suave.

Para entender el motivo por el cual las diferencias en los resultados son tan pequeñas, conviene en primer lugar tener claro, que el consumo de ACS representa un 10,8 % del consumo eléctrico total. De esta cantidad, una gran parte ya está cubierta por la generación de la instalación fotovoltaica. La producción de otra de las franjas horarias, aunque sea pequeña, no puede ser reasignada a otro momento del día dado que se tiene que dar la condición, de que el consumo se produzca más tarde que la generación. En el siguiente gráfico, se muestra demanda media diaria de ACS frente a la generación fotovoltaica diaria media.

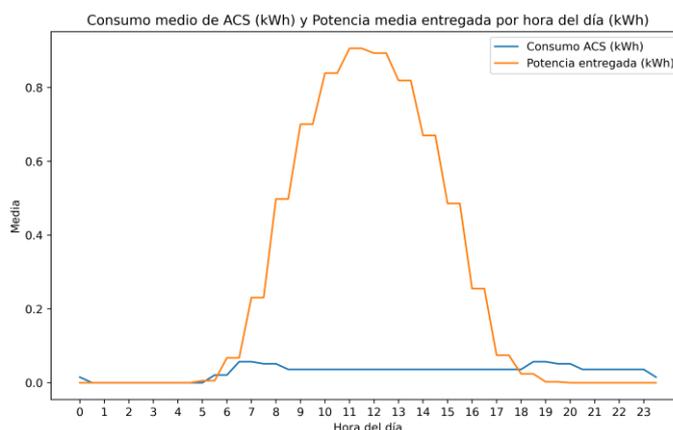


Figura 37: Consumo promedio de ACS frente a generación fotovoltaica promedio

Como se puede apreciar, y hablando en términos generales dado que se está tratando con datos promedio, solo una fracción de la producción, la que va desde aproximadamente las 19:00 horas en adelante, puede ser reasignada.

Dados los resultados mostrados, conviene, por tanto, seguir realizando análisis para diferentes casos y optimizando el funcionamiento del algoritmo. Por otro lado, es importante tener en cuenta que conforme el volumen de un depósito aumenta, sus pérdidas estáticas relativas, es decir, respecto del volumen y por tanto respecto de la energía almacenada en él, caen rápidamente. Este factor irá haciendo más conveniente una redistribución de la producción conforme el volumen de un depósito de ACS vaya aumentando.

En la siguiente sección, se presentan los resultados del enfoque de dimensionamiento conjunto. Para lograr esto, era necesario contar con más de una base de datos de consumo. Por lo tanto, se han generado dos bases de datos adicionales. Estas se han creado a partir de la base de datos original, multiplicando los datos del consumo eléctrico total por un factor específico y desplazándolos en el tiempo.

Casa	Nombre del archivo	Desplazamiento	Factor
Casa 2	consumo2.csv	1 h. hacia delante	1,05
Casa 3	consumo3.csv	1 h. hacia atrás	1,18

Tabla 16: Resumen de las Bases de Datos Creadas para la Simulación

A continuación, se presentan los resultados obtenidos.

```

----- RESULTADOS -----

Orientación elegida: 2
Nº ficha panel fotovoltaico: 10
Marca del panel fotovoltaico: Risen
Modelo del panel fotovoltaico: RSM110-8
Potencia del panel fotovoltaico: 535
Nº de módulos fotovoltaicos a instalar: 3
Nº de ramas en Serie: 3
Nº de ramas en paralelo: 0
Distancia a dejar entre modulos: 2.01 m.
Nº ficha inversor fotovoltaico: 6
Marca inversor fotovoltaico: Growatt
Modelo inversor fotovoltaico: MTL-S
Potencia inversor fotovoltaico: 2.5 kW.
Sección DC: 6.0 mm2
Caída de tensión en el tramo DC: 1.23 %
Sección AC: 1.5 mm2
Caída de tensión AC: 0.56 %
Coste de la instalación: 1321.54 €
Energía entregada: 4801 kWh/año
Energía aprovechada: 4279 kWh/año
% Energía aprovechada: 89.13 %
Rendimiento de la instalación: 75.8 %
Precio promedio de la energía de la instalación: 0.012352 €/kWh
Coste total de la instalación: 2856 €
Coeficientes de reparto: [0.3, 0.34, 0.35]
Coste de la instalación para la casa 1: 396.46 €
Precio de la energía para la casa 1 0.012382 €/kWh
Importe de la factura eléctrica para la casa 1: 876.11 €

```

Figura 38: Resultados – Dimensionado conjunto instalación fotovoltaica

Es posible advertir diversas diferencias que al igual que en el primer caso, la orientación elegida ha sido el número dos, que corresponde con una orientación de los paneles hacia el sur. El número de paneles, por su parte, ha aumentado en uno. Realmente, tanto en este resultado el número de paneles resultantes es asombrosamente bajo. Si bien la potencia de estos últimos supera los 500 W, la instalación de 1.605 W de potencia total para tres viviendas, es baja en comparación con las potencias que actualmente se instalan en las viviendas unifamiliares, donde, fácilmente se instalaría una potencia similar para cada una de las viviendas. Es importante recordar, que en esta simulación se está trabajando con consumos extraídos a su vez de un programa de simulación y que estos, aparte, han sido alterados de forma que se pudiera realizar la simulación. Las curvas de consumo resultantes son por tanto algo atípicas, lo cual influye en los resultados de la simulación. Sin embargo, es esencial destacar que esta peculiaridad no desacredita la simulación de ninguna manera; más bien, desde el punto de vista del autor, refuerza su utilidad. Las curvas de consumo inusuales podrían ser el resultado de estilos de vida poco comunes y, por lo tanto, deben evaluarse antes de considerar la implementación de una comunidad energética.

La marca, modelo y potencia del inversor, también ha variado, tal y como es lógico. En este punto es muy importante añadir, que el código actualmente no lleva ninguna instrucción que limite la potencia del inversor por arriba. Dado que la función 'selector\_principal' selecciona el módulo sin tener en cuenta información de los inversores, y atendiendo al criterio del precio más bajo, es posible encontrarse con paneles que más adelante dificulten la selección de un inversor. En este caso en concreto, la intensidad producida por el panel supera los diecisiete amperios, lo cual descarta la mayoría de los inversores en este rango de potencias. Para solucionar este problema existen básicamente dos soluciones sencillas: ampliar la base de datos de inversores y descartar paneles con estas características, bien de la base de datos de paneles o bien añadiendo un filtro al comienzo del código.

Por otra parte, es posible observar que la energía aprovechada ha aumentado en casi un 20%, lo cual es una muy buena noticia.

Finalmente, es posible advertir que tanto el precio del kilovatio-hora de la instalación como su precio disminuyen significativamente, no obstante, el de la factura eléctrica aumenta. Esto último, es debido a la cantidad de energía disponible para la vivienda evaluada, que, en este caso, es menor. La decisión resultante, dependerá de la inversión inicial que se quiera realizar, de la tenencia en cuenta del factor sostenibilidad, dado que una instalación con mayor producción de energía es más sostenible y de la cantidad de dinero que se pretenda ahorrar.

Se muestran, por último, los resultados de la ejecución del módulo de simulación.

```
----- RESULTADOS CASA SIMULACIÓN -----  
% Energía aprovechada total: 89.7 %  
Precio de la energía para la casa 1 0.012338 €/kWh  
Importe de la factura eléctrica para la casa 1: 874.88 €
```

Figura 39: Resultados – Modo Simulación

Al contrario de lo esperado, el precio del kilovatio-hora y el importe de la factura eléctrica apenas han descendido. Por su parte, la energía aprovechada por el conjunto de viviendas apenas ha aumentado. Dada la creencia de que el código contenía alguna clase de error, se procedió a un análisis para determinar qué estaba sucediendo. Tras este, se concluyó que no había ningún tipo

de fallo en el código, sino que, simplemente, dadas las características de los consumos eléctricos y las del dimensionado de la instalación, si bien existían numerosos momentos de déficit y excedente, no se originaban en todas las casas al mismo tiempo. De 17.520 datos de consumo eléctrico recogidos para cada una de las viviendas, apenas existían 50 en los cuales los periodos de déficit de una vivienda coincidieran con los de excedente de otra. Es necesario tener en cuenta, tanto la atipicidad en el consumo como que las bases de datos recogen datos cada periodo de media hora y que probablemente, una base de datos más extensa influya en los resultados de la simulación.

Es posible concluir, por tanto, que no es aconsejable formar una comunidad energética con este grupo de viviendas, o no exclusivamente con este grupo de viviendas. Será necesario buscar configuraciones diferentes o añadir potenciales miembros de la comunidad energética a la simulación, para evaluar su conveniencia.

## 6.9. Inyección de los Excedentes en la Red Eléctrica

El trabajo actual quedaría incompleto si no se considerara la opción de inyectar los excedentes de energía en la red eléctrica, una alternativa sin duda alguna sencilla y ventajosa. Esta posibilidad puede ser adoptada además de las otras opciones expuestas en el trabajo.

En el Real Decreto 244/2019, de 5 de abril, se encuentran las regulaciones referentes a las condiciones administrativas, técnicas y económicas del autoconsumo de energía eléctrica. En este sentido, se presentará a continuación una breve introducción a las distintas modalidades de autoconsumo habilitadas por dicho documento.

### 6.9.1. Modalidades

#### 6.9.1.1. Modalidad de suministro con autoconsumo sin excedentes

Esta modalidad, no contempla la inyección de excedentes en la red. Es necesario que la instalación cuente con un sistema antivertido según lo establecido en la ITC-BT-40.

#### 6.9.1.2. Modalidad de suministro con autoconsumo con excedentes

Contempla la inyección de energía excedentaria en las redes de transporte y distribución. Esta modalidad se divide a su vez en la modalidad con excedentes acogida a compensación y en modalidad con excedentes no acogida a compensación.

##### 6.9.1.2.1. Modalidad con excedentes acogida a compensación

La modalidad con excedentes acogida a compensación permite a los generadores de energía renovable inyectar el exceso de energía a la red eléctrica y compensar esta energía excedente con la energía consumida de la red en diferentes momentos dentro del mismo período de facturación (Un mes como máximo).

Para acogerse a esta modalidad, deben cumplirse los siguientes cinco requisitos:

- i. La fuente de energía primaria sea de origen renovable.
- ii. La potencia total de las instalaciones de producción asociadas no sea superior a 100 kW.
- iii. Si resultase necesario realizar un contrato de suministro para servicios auxiliares de producción, el consumidor haya suscrito un único contrato de suministro para el consumo asociado y para los consumos auxiliares de producción con una empresa comercializadora.

- iv. El consumidor y productor asociado hayan suscrito un contrato de compensación de excedentes de autoconsumo definido en el artículo 14 del Real Decreto 244/2019.
- v. La instalación de producción no tenga otorgado un régimen retributivo adicional o específico.

#### 6.9.1.2.2. Modalidad con excedentes no acogida a compensación:

Pertencen a esta modalidad los casos que no cumplan con los requisitos para pertenecer a la modalidad anterior o que voluntariamente hayan decidido no acogerse a ella. El productor, debe darse de alta en el Registro Administrativo de Instalaciones de Producción de Energía (RAIPEE) como productor de energías renovables. Los excedentes, se venderán en el mercado eléctrico.

#### 6.9.1. Mecanismo de compensación simplificada

El mecanismo de compensación simplificada consistirá en un saldo en términos económicos de la energía consumida en el periodo de facturación con las siguientes características:

- En el caso de que se disponga de un contrato de suministro con una comercializadora libre:
  - o La energía horaria consumida de la red será valorada al precio horario acordado entre las partes.
  - o La energía horaria excedentaria, será valorada al precio horario acordado entre las partes.
- En el caso de que se disponga de un contrato de suministro al precio voluntario para el pequeño consumidor con una comercializadora de referencia:
  - o La energía horaria consumida de la red será valorada al coste horario de energía del precio voluntario para el pequeño consumidor en cada hora, TCUh, definido en el artículo 7 del Real Decreto 216/2014, de 28 de marzo.
  - o La energía horaria excedentaria, será valorada al precio medio horario, Pmh; obtenido a partir de los resultados del mercado diario e intradiario en la hora h, menos el coste de los desvíos CDSVh, definidos en los artículos 10 y 11 respectivamente del Real Decreto 216/2014, de 28 de marzo.

Por otra parte, la energía horaria excedentaria de los consumidores acogidos al mecanismo de compensación simplificada no tendrá consideración de energía incorporada al sistema eléctrico de energía eléctrica y, en consecuencia, estará exenta de satisfacer los peajes de acceso.

#### 6.10. Estimación del importe ahorrado

Dadas las características de la instalación, parece razonable optar por una modalidad con excedentes acogida a compensación. Tal y como se ha comentado en párrafos anteriores, el importe de la factura eléctrica dependerá tanto del tipo de contrato que se suscriba y, dado el caso, de las negociaciones que se lleven a cabo. No obstante, de cara a tener un marco de referencia, se hará una estimación para un contrato de suministro al precio voluntario para el precio del consumidor.

En el siguiente gráfico, se muestran los valores del precio del megavatio-hora, tanto para el término de facturación de energía excedentaria para el mecanismo de compensación simplificada (PVPC) como para el término de facturación de energía activa del PVPC, ambos para la península.

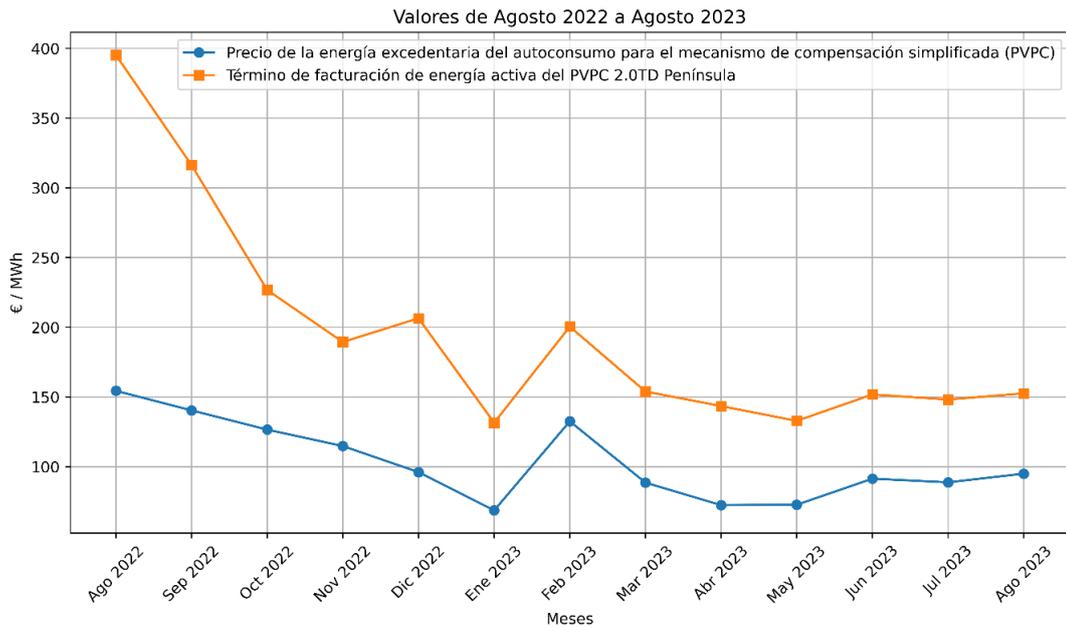


Figura 40: Precio de la energía excedentaria frente al término de facturación de energía activa (PVPC)

Durante el período de análisis, en promedio, la energía consumida de la red ha tenido un costo un 49 % mayor que la energía excedentaria volcada a la red. Utilizando este dato y considerando los excedentes energéticos durante un año, se obtiene el siguiente desglose:

Mes	Excedente (kWh)	Descuento factura (€)
Enero	3,61	0,55
Febrero	24,07	3,68
Marzo	43,64	6,68
Abril	49,54	7,58
Mayo	84,54	12,93
Junio	96,50	14,76
Julio	93,92	14,37
Agosto	96,42	14,75
Septiembre	99,59	15,24
Octubre	76,88	11,76
Noviembre	19,61	3,00
Diciembre	3,61	0,55
<b>Total</b>	<b>691,93</b>	<b>105,86</b>

Tabla 17: Excedentes por mes y descuento en la factura eléctrica

El período de amortización, calculado como la inversión dividida por el ahorro anual, resulta en:

$$\text{Periodo de amortización (Años)} = \frac{\text{Inversión (€)}}{\text{Ahorro anual}}$$

$$\text{Periodo de amortización (Años)} = \frac{815,46}{661,96} = 1,23$$

## 7. Empleo de técnicas de Machine Learning para la predicción de la potencia generada en plantas fotovoltaicas

### 7.1. Introducción

El aprendizaje automático es una disciplina de la inteligencia artificial que permite a las máquinas aprender de los datos para mejorar su rendimiento en la realización de tareas, sin ser explícitamente programadas. (Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development.)

Existen multitud de tipos de sistemas de Machine Learning, los cuales pueden agruparse en categorías en función de diversos criterios, en función de si aprenden o no bajo la supervisión humana, de si se tienen o no las etiquetas (respuestas correctas), de si pueden aprender de forma gradual etc.

Este capítulo se pretende hacer una pequeña aproximación a los algoritmos más simples de este campo, sin entrar en demasiado detalle, y utilizar la biblioteca "Scikit-Learn", de Python, para entrenarlos en el conjunto de datos disponible y hacer algunas predicciones.

#### Definición

Regresión: Es una técnica estadística utilizada para modelar la relación entre una variable dependiente y otras independientes.

### 7.2. Algoritmos de regresión

#### 7.2.1. Regresión lineal

Es el modelo más sencillo de todos. Con él, se trata de encontrar la línea recta que mejor se ajuste a los datos disponibles.

La ecuación de regresión lineal tiene la forma:

$$\hat{Y} = \hat{\beta}_0 + \sum_{j=1}^p X_j \hat{\beta}_j$$

*Ecuación 13: Regresión lineal – Forma explícita*

Donde:

- $\hat{Y}$ : Es la variable dependiente predicha
- $\hat{\beta}_0$ : Es el término de intersección
- $X_j$ : Es el valor de la j-ésima variable independiente.
- $\hat{\beta}_j$ : Es el coeficiente de la j-ésima variable independiente.

Expresada de forma matricial:

$$\hat{Y} = X\hat{\beta}$$

*Ecuación 14: Regresión lineal – Forma matricial*

La RSS (Residual Sum of Squares) es una medida utilizada en estadística para cuantificar la discrepancia entre los valores reales y los predichos por el modelo.

$$RSS(\beta) = \sum_{i=1}^N (y_i - x_i^T \beta)^2$$

*Ecuación 15: Suma de los cuadrados de los residuos – Forma explícita*

Expresada de forma matricial:

$$RSS(\beta) = (y - X\beta)^T (y - X\beta)$$

*Ecuación 16: Suma de los cuadrados de los residuos – Forma Matricial*

Para minimizar la RSS, es necesario encontrar los valores  $\beta$  que la minimizan. Derivando la expresión anterior respecto de  $\beta$  e igualándola a cero se obtiene:

$$X^T (y - X\beta) = 0$$

Por último, despejando  $\beta$  se obtiene:

$$\hat{\beta} = (X^T X)^{-1} X^T y$$

En las siguientes gráficas se pueden apreciar un conjunto de datos lineales y otro de datos no lineales. Para ambos, se ha obtenido la recta que minimiza la RSS. Tal y como se puede apreciar, mediante la recta obtenida para el primer modelo se podrá predecir con bastante exactitud la variable dependiente. La recta obtenida para el segundo modelo hará predicciones aceptables para un rango de valores X de entre -4 y 7, no obstante, cometerá un error mucho mayor conforme los datos se salgan de este rango.

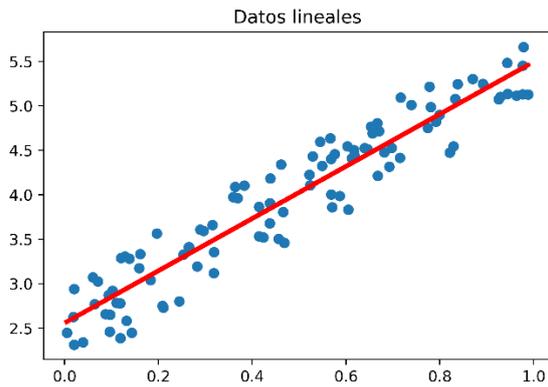


Figura 41: Ajuste de regresión lineal en datos lineales

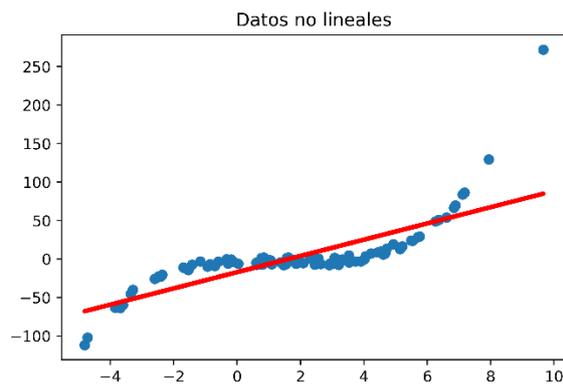


Figura 42: Ajuste de regresión lineal en datos no lineales

### 7.2.2. Regresión polinomial

Es una extensión de la regresión lineal en la que se ajusta una función polinomial a los datos en lugar de una línea recta.

En la práctica, se realiza una transformación polinomial a los datos, para posteriormente aplicarles un modelo de regresión lineal.

A continuación, se muestra una tabla con dos atributos (a y b), con 3 observaciones para cada uno de ellos.

Observaciones	<i>a</i>	<i>b</i>
1	1	-2
2	2	3
3	4	1

Tabla 18: Valores observados de los atributos a y b

Para realizar la transformación polinomial, se genera una nueva matriz de características con todas las combinaciones polinomiales de las características de grado menor o igual al especificado.

Para, por ejemplo, un polinomio de grado 2, la matriz resultante estaría formada por una constante (Término independiente del modelo), los valores originales, sus cuadrados y el producto entre los valores originales.

Constante	Lineal		Cuadrática		Interacción
-	<i>a</i>	<i>b</i>	<i>a</i> <sup>2</sup>	<i>b</i> <sup>2</sup>	<i>a * b</i>
1	1	-2	1	4	-2
1	2	3	4	9	6
1	4	1	16	1	4

Tabla 19: Transformación polinomial de los atributos a y b

Estos términos se utilizarían para ajustar el modelo de regresión, resultando en una ecuación como esta:

$$y = c + a_1a + b_1b + a_2a^2 + b_2b^2 + (ab)_1ab$$

Ecuación 17: Expresión de un modelo de regresión polinomial de segundo grado con término de interacción

En la siguiente gráfica se puede observar un polinomio de segundo grado, ajustado a un conjunto de datos que sigue una distribución polinomial.

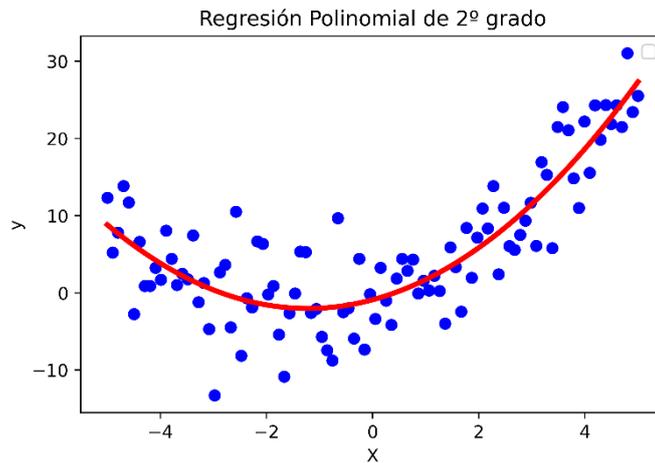


Figura 43: Ajuste de regresión polinomial de 2º grado a datos con distribución polinomial de 2º grado y ruido

### 7.2.3. Árboles de decisión

Constituyen un enfoque fundamental dentro del llamado “aprendizaje supervisado”. Se denominan así debido a que pueden representarse visualmente como una estructura con forma de árbol. En esta representación, cada nodo interno simboliza una característica o atributo, cada arista con una regla de decisión y cada nodo-hoja con una etiqueta.

A continuación de muestra un ejemplo muy simple de un árbol de decisión creado para predecir el precio de una vivienda.

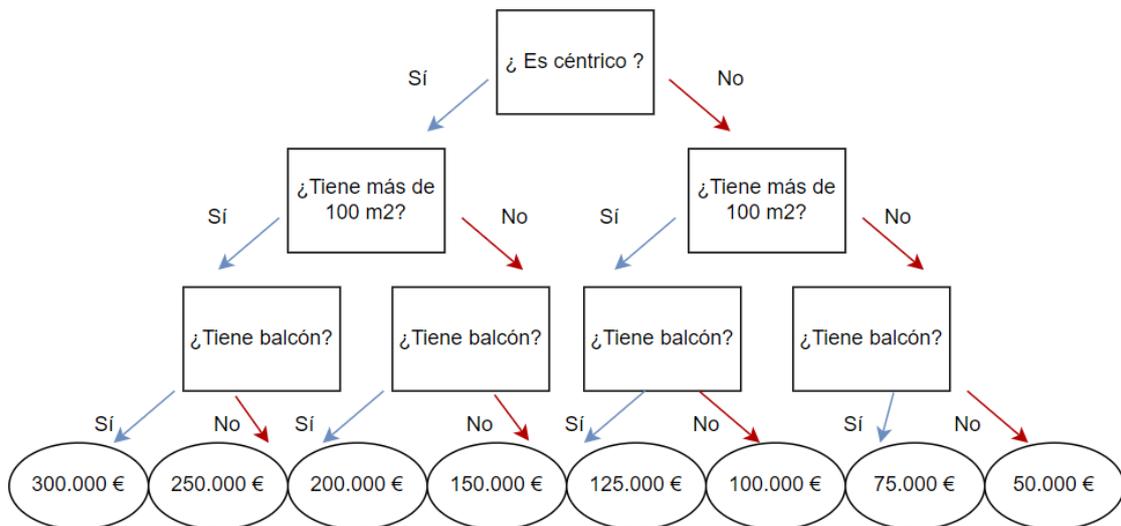


Figura 44: Ilustración simplificada de un árbol de decisión para la estimación del precio de una vivienda

Como comentado anteriormente, cada árbol se compone de varios elementos: nodos, ramas y hojas.

- Nodos: Son los puntos de decisión del árbol y representan una pregunta o prueba. En este caso, los nodos son "¿Es céntrico?", "¿Tiene más de 100 m<sup>2</sup>?" y "¿Tiene terraza?".
- Ramas: Representan la solución a la pregunta hecha en el nodo. Cada nodo tiene dos o más ramas que representan las posibles respuestas a la pregunta del nodo. En este caso, las ramas representan las respuestas "Sí" o "No".
- Hojas: Representan los resultados finales o las decisiones tomadas después de recorrer los nodos y las ramas. En este caso, el precio de la vivienda.

Cada vez que se hace una pregunta, el árbol se bifurca en tantas ramas como respuestas haya para la pregunta. Estas ramas conducen después a otros nodos, o a hojas, donde se toma la decisión final.

Un concepto importante es el de entropía o el índice de Gini, que son medidas de impureza o incertidumbre. Cuando se construye un árbol de decisión, se intenta hacer preguntas que reduzcan todo lo posible la impureza o incertidumbre de las respuestas. Si una pregunta divide perfectamente los datos, se dice que tiene una entropía o índice de Gini de cero. De igual manera, si una pregunta divide los datos de una manera que hay mucha impureza, se dice que tiene una entropía o índice de Gini alto.

### 7.3. Métricas de validación

A continuación, se exponen algunas de las métricas más comunes, utilizadas para evaluar la precisión de los modelos entrenados.

- RMSE (Root Mean Square Error): Es la raíz cuadrada de la media, de las diferencias entre los valores reales y los valores predichos, elevadas al cuadrado. Penaliza grandes errores en las predicciones.
- MAPE (Mean Absolute Percentage Error): Mide la desviación absoluta media entre el valor real y el predicho en términos de porcentaje. Es útil para contextos en los que el error relativo es más importante que la magnitud absoluta del error.
- $R^2$  (Coeficiente de determinación): Proporciona una medida sobre cuán bien las predicciones del modelo se ajustan a los valores reales. Es útil para comparar modelos y para evaluar qué proporción de la variabilidad de los datos puede ser explicada por el modelo.

### 7.4. Sobreajuste y subajuste

Por una parte, existen modelos, que no tienen la capacidad de ajustarse suficientemente bien a los datos. Por otra parte, existen modelos que se ajustan demasiado al conjunto de datos. En el primer caso se habla de subajuste; en el segundo, de sobreajuste.

El problema del primer caso es evidente, un modelo subajustado no será, probablemente, lo suficientemente preciso.

El problema del segundo es que funcionan muy bien en el conjunto de datos de entrenamiento, sin embargo, no son capaces de generalizar adecuadamente a otros conjuntos de datos, pudiendo cometer errores muy grandes. Por otra parte, pueden aprender a reproducir parte del ruido del conjunto, perjudicando su rendimiento en nuevos conjuntos.

Esto, puede observarse en los gráficos siguientes. A la izquierda se puede apreciar un modelo subajustado, en el centro un modelo bien ajustado, y a la derecha un modelo sobreajustado.

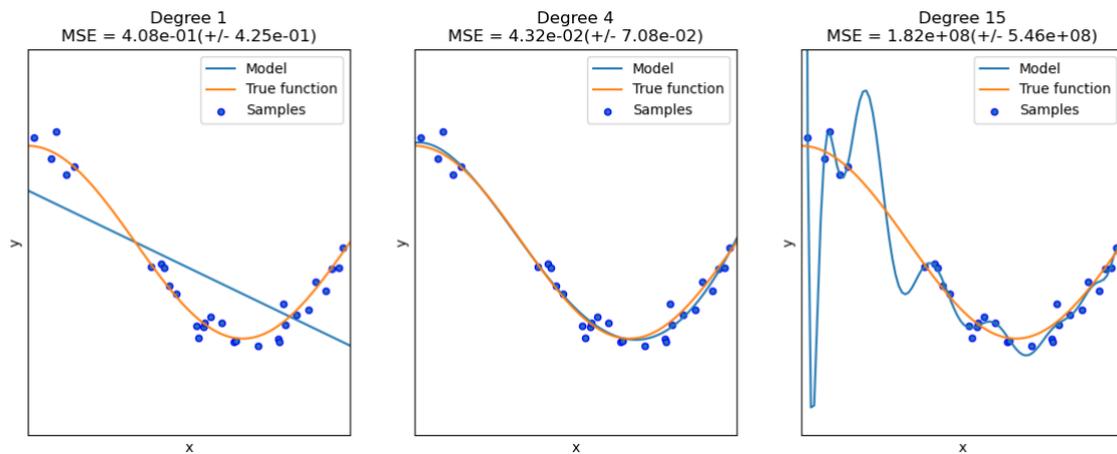


Figura 45: Subajuste, ajuste correcto y sobreajuste en modelos predictivos. Fuente: [https://scikit-learn.org/stable/auto\\_examples/model\\_selection/plot\\_underfitting\\_overfitting.html](https://scikit-learn.org/stable/auto_examples/model_selection/plot_underfitting_overfitting.html)

## 7.5. Proyecto

La instalación sujeta de estudio es una planta fotovoltaica de aproximadamente 25 kilovatios, situada en Paterna, Valencia. Se ha escogido esta ubicación, dada la necesidad de contar con bases de datos lo suficientemente amplias, con registros no solo de la potencia generada de la instalación sino también, de otras variables influyentes sobre la generación. Si bien la ubicación es diferente a la ubicación de la vivienda, la secuencia de pasos a seguir, una vez recopilados suficientes datos pueden ser idéntica. Aparte, muchas de las conclusiones como la validez de estos algoritmos para este tipo de aplicación o sobre la influencia de las diferentes variables sobre la producción energética son extrapolables y ayudan a una mayor comprensión, esté donde esté situada la instalación analizada.

A continuación, se presenta un cuadro con algunos de los datos disponibles.

<b>Datos</b>	<b>Valor</b>
<b>Localización</b>	-
Latitud (°)	39,53
Longitud (°)	0,440599
Ángulo de inclinación del panel (°)	15
Ángulo azimutal del panel(°)	0
<b>Panel</b>	-

Coefficiente de temperatura (% / °C)	-0,0037
Potencia nominal (W)	315
NOTC (°C)	44,6
<b>Instalación</b>	-
Número de módulos	76
Potencia nominal (W)	23940

Tabla 20: Características de la planta fotovoltaica bajo estudio

Por otra parte, se dispone de una base de datos con 448.939 entradas, correspondientes a valores medidos a lo largo del año 2021 en intervalos de un minuto. Esta base de datos ha sido preprocesada previamente, de manera que pueda utilizarse para el aprendizaje de los algoritmos. Durante la limpieza de la base de datos se han sustraído filas en las cuales faltaba algún valor y se han eliminado filas con valores claramente atípicos, como pueden ser potencias negativas.

A continuación, se presentan las características incluidas en la base de datos.

Atributo	Comentario
t	Fecha y hora
Presión (Bar)	-
Dirección Viento (°)	-
Velocidad Viento (m/s)	-
Humedad (%)	Humedad relativa
Radiación (W / m <sup>2</sup> )	Radiación sobre plano horizontal
Temperatura (°C)	-
Potencia (W)	Potencia activa entregada por la instalación
Radiación corregida (W / m <sup>2</sup> )	Radiación corregida para plano inclinado

Tabla 21: Características de la base de datos de la planta fotovoltaica

El punto de partida es el de un modelo que predice la potencia entregada basándose en ecuaciones físicas como las descritas en capítulos anteriores.

A continuación, se presentan cuatro gráficas en las cuales puede apreciarse la comparación entre la potencia real y la potencia del modelo previamente descrito.

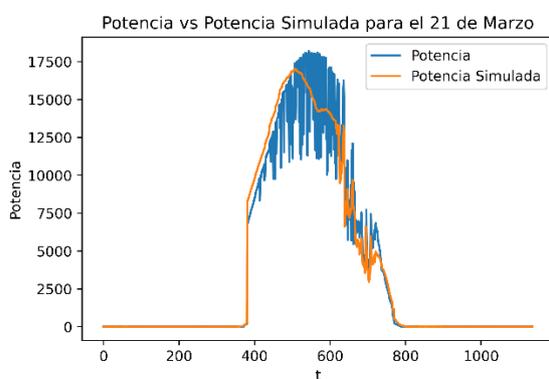


Figura 46: Potencia real frente a potencia estimada por el modelo inicial para el 21 de marzo

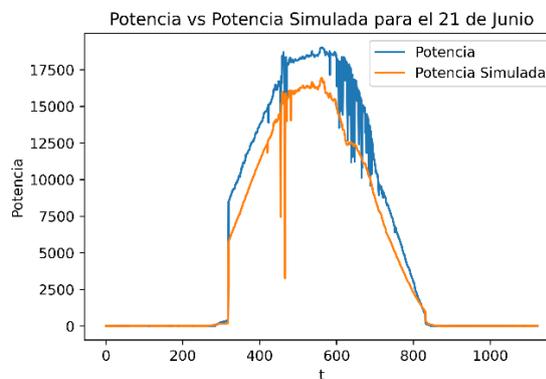


Figura 47: Potencia real frente a potencia estimada por el modelo inicial para el 21 de junio

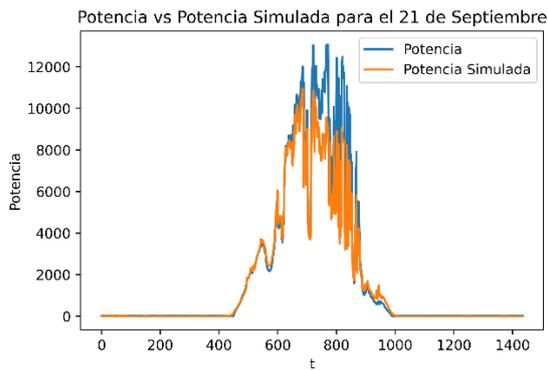


Figura 48: Potencia real frente a potencia estimada por el modelo inicial para el 21 de septiembre

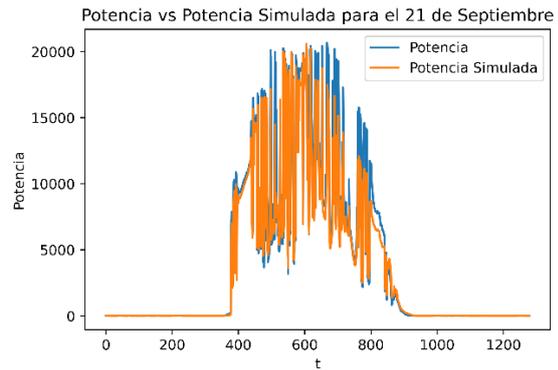


Figura 49: Potencia real frente a potencia estimada por el modelo inicial para el 21 de septiembre

Las métricas de evaluación de este modelo son las siguientes.

Métrica	Valor
RMSE	1121,77
MAPE	60,92
R <sup>2</sup>	0,95

Tabla 22: Métricas de evaluación – Modelo inicial

Para el modelo de Machine Learning, se ha descompuesto la columna con la fecha 't' en tres columnas: Mes, Hora y Minuto. A su vez, se han añadido los siguientes atributos:

- P\_sim: Potencia simulada mediante el modelo basado en ecuaciones físicas.
- omega: Ángulo horario solar.
- delta: Declinación solar.
- Azs: Ángulo azimutal solar
- theta: Ángulo de incidencia solar. Es el ángulo subtendido entre los rayos de sol entrantes y la normal a la superficie del panel.

En la siguiente figura se presentan diecisiete histogramas, que representan la distribución de los datos para cada uno de los atributos.

Hay modelos de Machine Learning que asumen una distribución normal en los datos. Puede ser necesario pues, procesarlos y transformarlos para que se ajusten a ese tipo de distribución. Además, los histogramas permiten identificar de forma visual valores atípicos que tengan que ser removidos de forma previa al entrenamiento. En este caso, todos los datos parecen caer dentro de un rango aceptable. Sin embargo, es observable que una gran proporción de los datos de radiación y, en consecuencia, de potencia, se concentran entorno a cero. Esto, es comprensible dado que durante aproximadamente la mitad de las horas del día no hay luz solar. No obstante, estos datos podrían no proporcionar información significativa para los modelos.

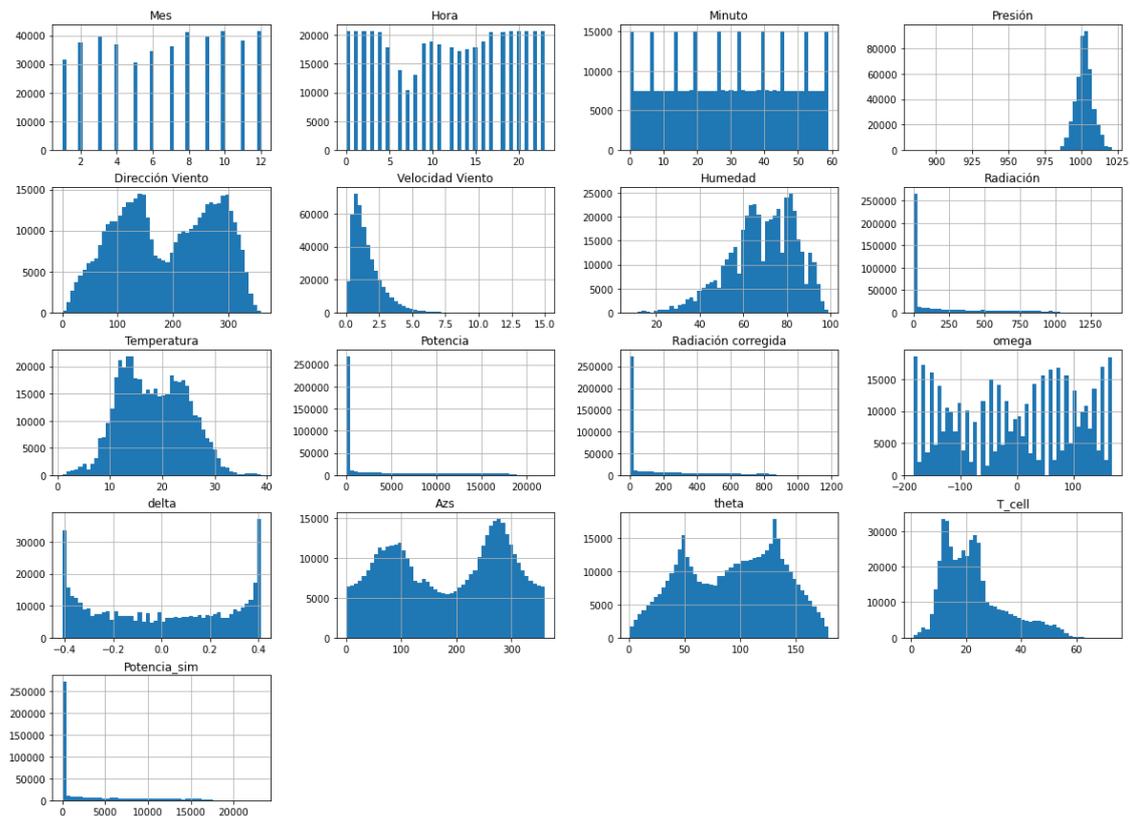


Figura 50: Histogramas de la distribución de los atributos

Existen diversas técnicas para identificar la correlación entre las características de un conjunto de datos y la variable objetivo. Dos de las más comunes son los coeficientes de correlación de Pearson y de Spearman. El primero, mide la relación lineal entre las variables mientras que el segundo, la relación no lineal. Ambos coeficientes varían en un rango de -1 a 1, con -1 indicando una fuerte correlación negativa y 1 representando una fuerte correlación positiva.

Seguidamente, se exhibe una tabla con los coeficientes mencionados, ordenados de mayor a menor.

<i>Atributo</i>	<i>Coefficientes</i>		<i>Atributo</i>
-	<b>Pearson</b>	<b>Spearman</b>	-
Potencia	1	1	Potencia
Potencia_sim	0,984691	0,842241	Radiación
Radiación corregida	0,983807	0,839251	Radiación corregida
Radiación	0,980654	0,838956	Potencia_sim
T_cell	0,866567	0,700932	T_cell
Temperatura	0,466707	0,435929	Temperatura
Velocidad Viento	0,365867	0,35671	Velocidad Viento
delta	0,159824	0,1794	delta
Azs	0,03492	0,039533	Mes
Hora	0,028648	0,017391	Minuto
omega	0,028548	-0,010168	Dirección Viento
Mes	0,016097	-0,051005	Presión
Minuto	0,003134	-0,077955	Azs

Dirección Viento	-0,006049	-0,088736	omega
Presión	-0,036015	-0,088736	Hora
Humedad	-0,518662	-0,434597	Humedad
theha	-0,775522	-0,739849	theha

Tabla 23: Coeficientes de correlación

Como es lógico, las correlaciones más fuertes se dan con las variables de potencia simulada y radiación, seguidos por el de temperatura de celda. Es posible agrupar las características en las siguientes categorías:

- Correlación muy fuerte: 'P\_sim', 'Radiación', 'Radiación corregida' y 'T\_cell'.
- Correlación fuerte: 'theta', 'Temperatura' y 'Humedad'.
- Correlación moderada: 'Velocidad Viento'.
- Correlación débil: 'delta'.
- Sin correlación: 'Azs', 'Hora', 'Mes', 'Minuto', 'omega' 'Dirección Viento', y 'Presión'

Existen variables, que pueden tener correlación con la potencia debido al efecto sobre otra de las variables. En este caso, la variable 'Humedad', puede tener efecto sobre la variable 'Radiación', debida a la dispersión de los rayos del sol debida a las gotas de agua presentes en el aire. No obstante, puede estar también implicada en la disipación de calor por medio de la evaporación en la superficie de las placas.

Variables relacionadas con la posición del sol, como 'theta' y 'delta' pueden estar correlacionadas con la potencia dada su relación con la radiación. Sin embargo, estas variables también podrían contener información sobre el impacto de las sombras sobre la producción de energía de la instalación.

Por su parte, la variable 'Velocidad Viento' podría estar correlacionada con la potencia entregada debido a la disipación de calor por el efecto de convección.

En los gráficos que se muestran a continuación, se muestran los atributos más relevantes trazados entre ellos. Pese a tener información cuantitativa sobre las correlaciones, en muchos casos puede ser útil representarlas de forma visual. Como señala Aurelien Geron (2020) en su libro 'Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow', nuestro cerebro es excepcionalmente bueno detectando patrones en imágenes.

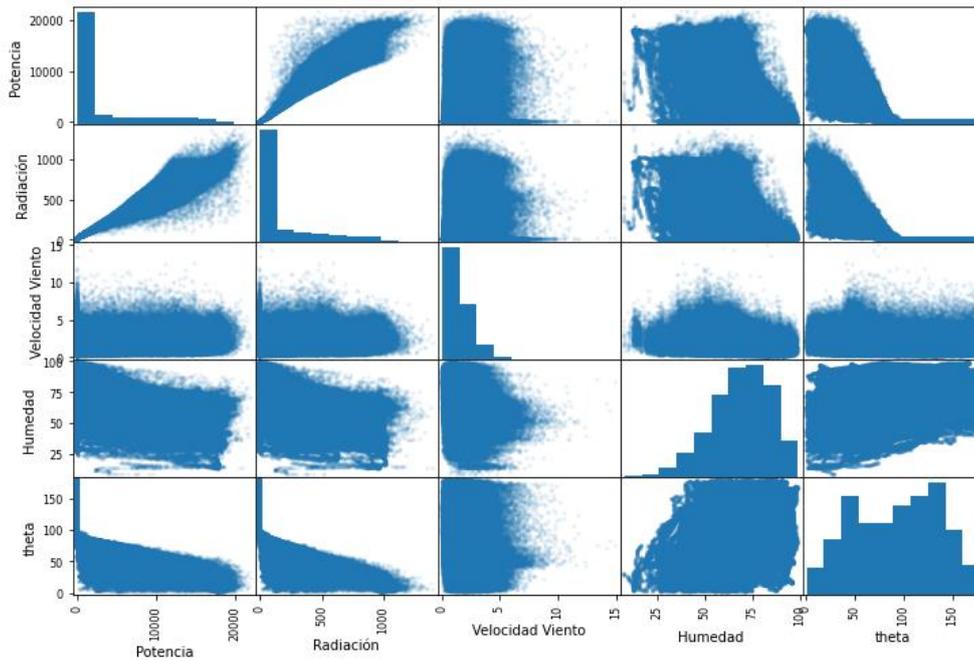


Figura 51: Gráficos de correlación

Es importante señalar, que existen modelos muy potentes, como pueden ser las redes neuronales, que pueden detectar patrones imperceptibles por los métodos mostrados. No obstante, para este proyecto, dados los algoritmos que se van a emplear, los atributos categorizados previamente como de 'Correlación débil' y 'Correlación moderada' serán eliminados.

Previo a esto, el conjunto de datos es segmentado en un dos: uno de entrenamiento y otro de prueba. El conjunto de prueba contiene un 5 % del total de los datos, seleccionados al azar. Este conjunto se reserva exclusivamente para la evaluación de los algoritmos más prometedores. El objetivo de este paso es permitir que los modelos sean probados en un conjunto de datos que no hayan 'visto' durante su aprendizaje, proporcionando así un medio para evaluar cualquier posible sobreajuste.

La siguiente tabla muestra los resultados obtenidos para los algoritmos de regresión lineal y polinomial.

Regresión lineal		Regresión polinomial - grado 4	
RMSE	851,15	RMSE	677,58
MAPE	471,67	MAPE	98,14
R <sup>2</sup>	0,97	R <sup>2</sup>	0,983
Regresión polinomial - grado 2		Regresión polinomial - grado 5	
RMSE	764,88	RMSE	664,21
MAPE	260,63	MAPE	93,98
R <sup>2</sup>	0,979	R <sup>2</sup>	0,9838
Regresión polinomial - grado 3		Regresión polinomial - grado 6	
RMSE	713,36	RMSE	657,04
MAPE	164,24	MAPE	108,95
R <sup>2</sup>	0,982	R <sup>2</sup>	0,984

Tabla 24: Métricas de evaluación para los modelos de regresión lineal y polinomial

Es importante aclarar, que la regresión polinomial de grado seis ha sido entrenada con un 30% de los datos disponibles en el conjunto de entrenamiento, debido a necesidad de mantener un equilibrio entre la exigencia computacional y los recursos disponibles. Aunque la intención inicial era la de emplear la totalidad del conjunto, la memoria RAM del equipo utilizado no tiene la capacidad suficiente como para manejar un volumen de datos tan grande como el requerido en este modelo.

Dado que los polinomios de alto grado son propensos a sobreajustarse a los datos de entrenamiento, resulta útil emplear técnicas de validación cruzada. En ellas, se segmenta en conjunto de entrenamiento en varias particiones. Cada partición se utiliza una vez como conjunto de prueba, mientras que el resto es utilizado para entrenar el modelo. Este proceso se realiza hasta que todas las particiones han sido utilizadas una vez como conjunto de prueba.

En la siguiente tabla se muestra el resultado del entrenamiento mediante validación cruzada para la regresión polinomial de grado cinco. Dada la similitud de los resultados, es posible concluir que no existe sobreajustamiento en el modelo. Dado que este es el polinomio de mayor grado entrenado con el conjunto completo, se concluye que los demás modelos tampoco han sobreajustado.

N.º	RMSE
1	687,99
2	637,5
3	685,9
4	695,48
5	654,65
6	637,51
7	660
8	726,54
9	1441,53
10	668,73
Media	749,58
Desviación estándar	232,11

Tabla 25: Datos sobre el RMSE obtenido mediante validación cruzada para la regresión polinomial de grado 5

La evaluación del algoritmo del tipo árbol de decisión, arroja por su parte los siguientes resultados.

Árbol de decisión	
RMSE	667,605
MAPE	5,2
R <sup>2</sup>	0,984

Tabla 26: Métricas de evaluación el modelo tipo "Decision Tree"

Es posible apreciar una ligera disminución de la RMSE respecto de los algoritmos polinomiales, y una importante reducción del MAPE.

No obstante, tal y como se muestra a continuación, mediante la técnica de validación cruzada el RMSE obtenido es casi doscientos puntos mayor. El modelo, se ha sobreajustado ligeramente a los datos.

Nº	RMSE
1	844,83
2	836
3	825,69
4	845,41
5	848,22
6	803,92
7	823,06
8	838,09
9	800,89
10	830,07
Media	829,62
Desviación estándar	15,76

Tabla 27: Datos sobre el RMSE obtenido mediante validación cruzada para el modelo tipo "Decision Tree"

Pese a haber detectado un ligero sobreajuste en el algoritmo anterior, el MAPE obtenido ha sido mucho menor al del resto de los algoritmos. Por tanto, se ha decidido utilizar tanto este modelo como el del polinomio de grado cinco para la prueba en el conjunto de entrenamiento.

## 7.6. Resultados

A continuación, se muestran los resultados, acompañados de dieciséis gráficas, de forma que se pueda comparar lo mejor posible el modelo actual con el modelo inicial.

<b>Métricas de evaluación en el conjunto de prueba</b>			
<b>Regresión polinomial - grado 5</b>		<b>Árbol de decisión</b>	
RMSE	654,24	RMSE	822,71
MAPE	90,8	MAPE	3,95
R <sup>2</sup>	0,9848	R <sup>2</sup>	0,976

Tabla 28: Métricas de validación sobre el conjunto de prueba para los modelos con mejores resultados

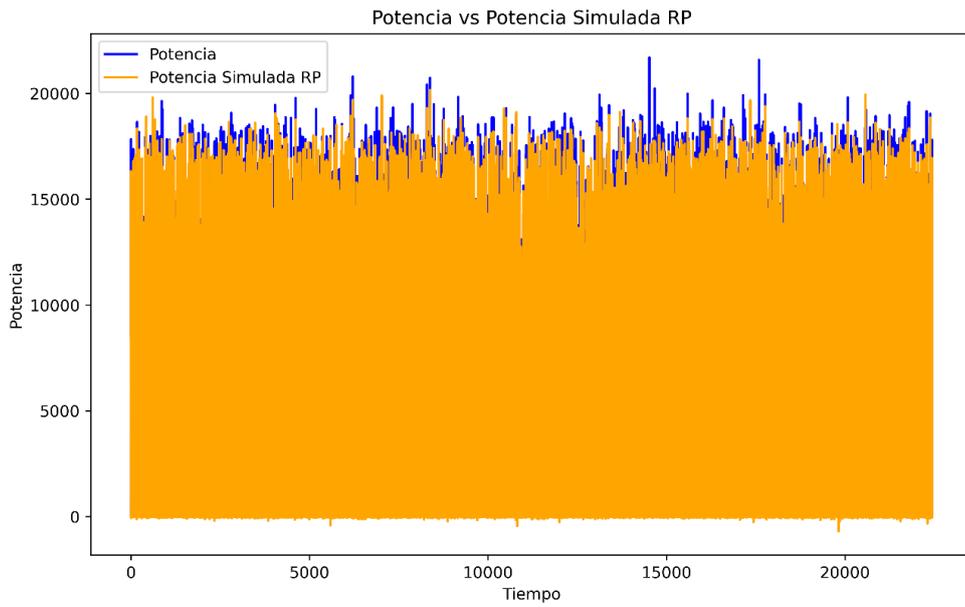


Figura 52: Potencia real frente a potencia simulada mediante el modelo de regresión polinomial

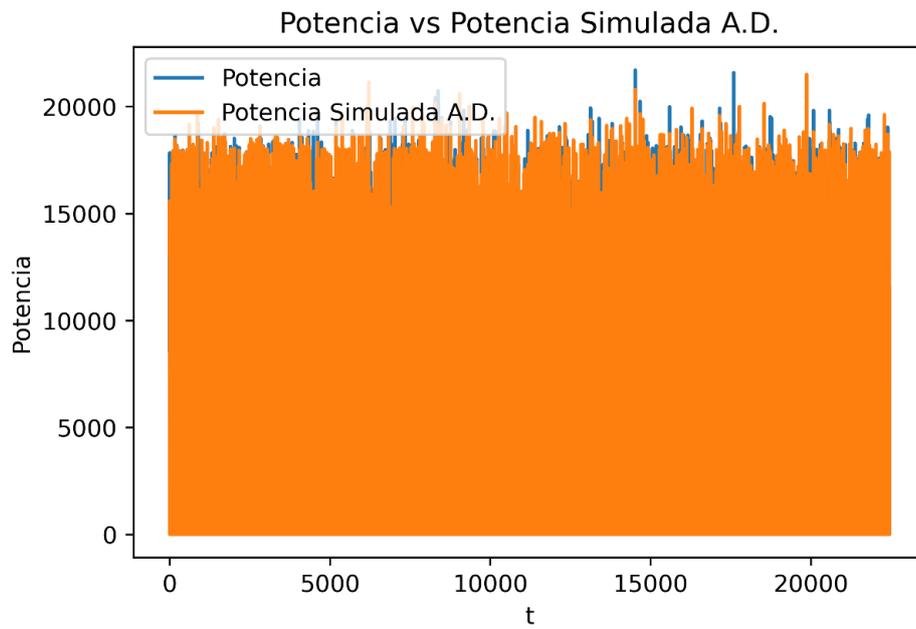


Figura 53: Potencia real frente a potencia simulada mediante el modelo tipo "Decision Tree"

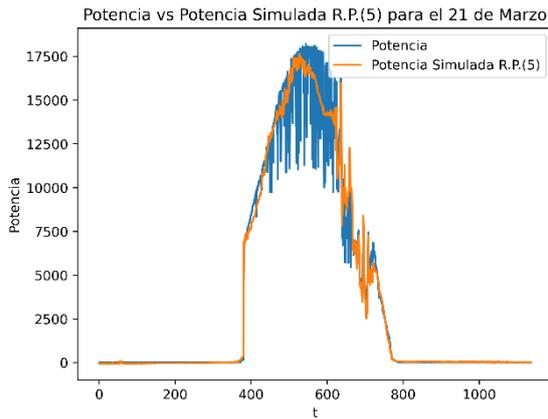


Figura 54: Potencia real frente a potencia estimada por el modelo de regresión polinomial para el 21 de marzo

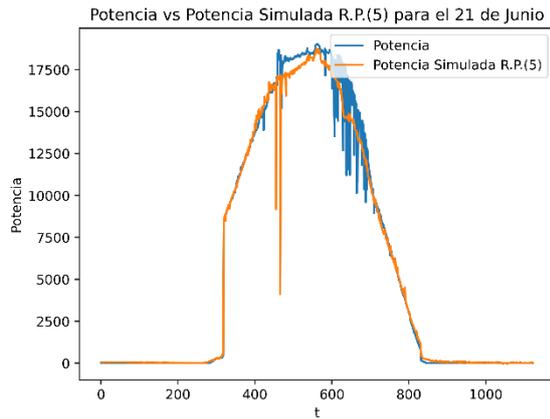


Figura 55: Potencia real frente a potencia estimada por el modelo de regresión polinomial para el 21 de junio

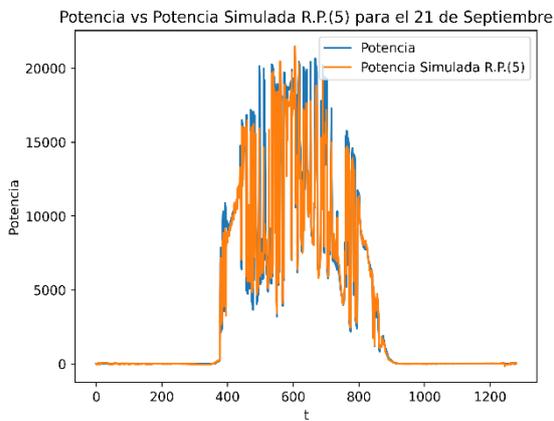


Figura 56: Potencia real frente a potencia estimada por el modelo de regresión polinomial para el 21 de septiembre

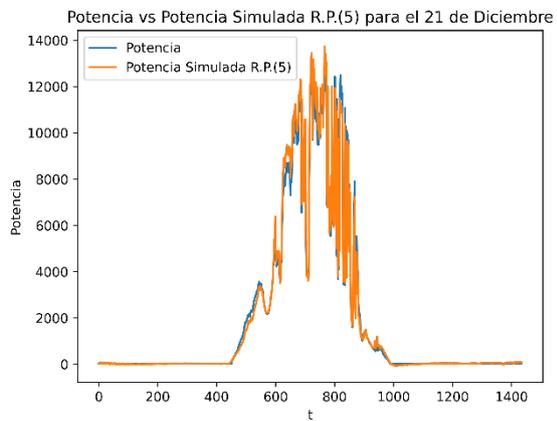


Figura 57: Potencia real frente a potencia estimada por el modelo de regresión polinomial para el 21 de diciembre

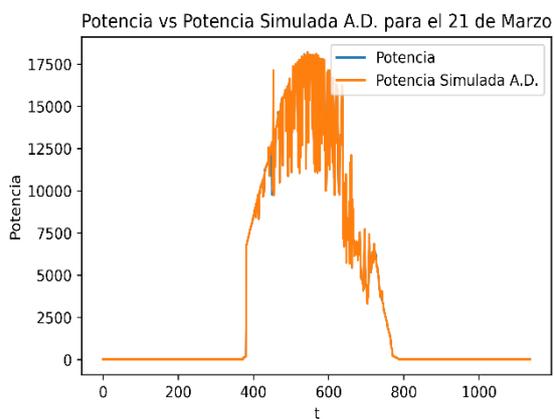


Figura 58: Potencia real frente a potencia estimada por el modelo de árbol de decisión para el 21 de marzo

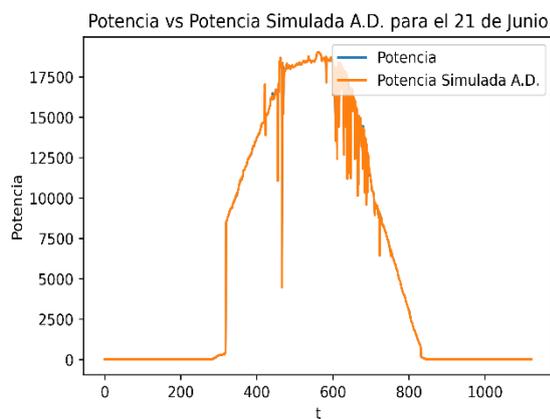


Figura 59: Potencia real frente a potencia estimada por el modelo de árbol de decisión para el 21 de junio

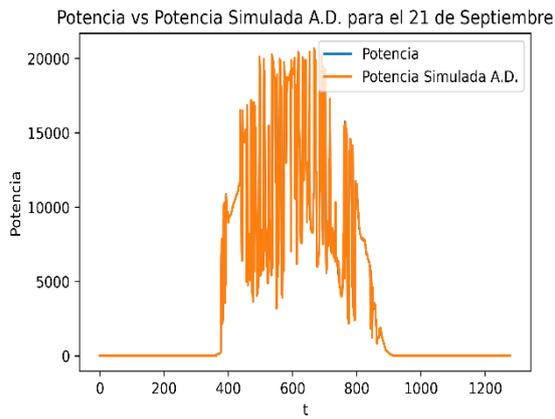


Figura 60: Potencia real frente a potencia estimada por el modelo de árbol de decisión para el 21 de septiembre

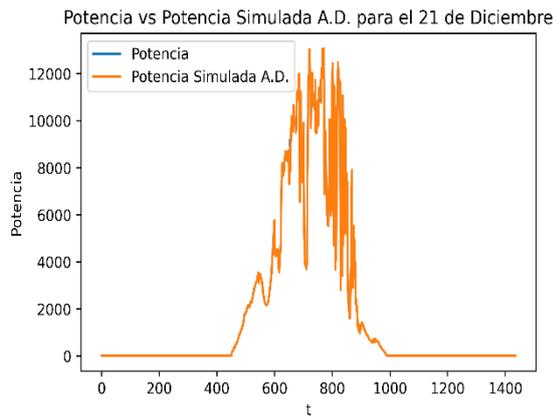


Figura 61: Potencia real frente a potencia estimada por el modelo de árbol de decisión para el 21 de diciembre

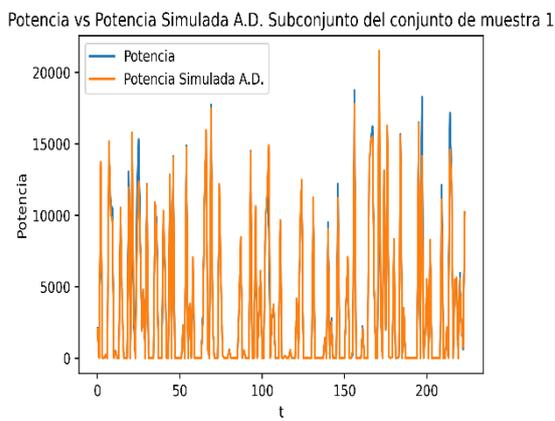


Figura 62: Potencia real frente a potencia estimada por el modelo de árbol de decisión para el subconjunto 1

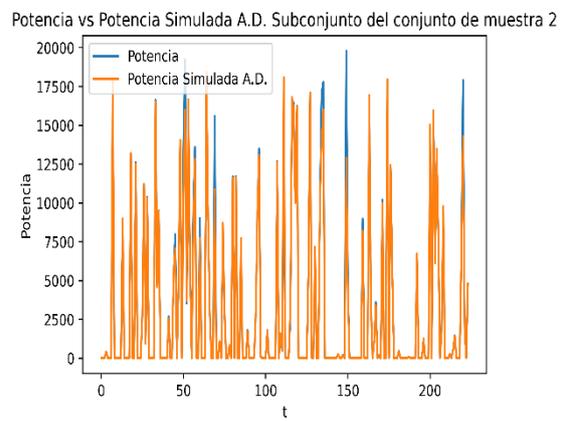


Figura 63: Potencia real frente a potencia estimada por el modelo de árbol de decisión para el subconjunto 2

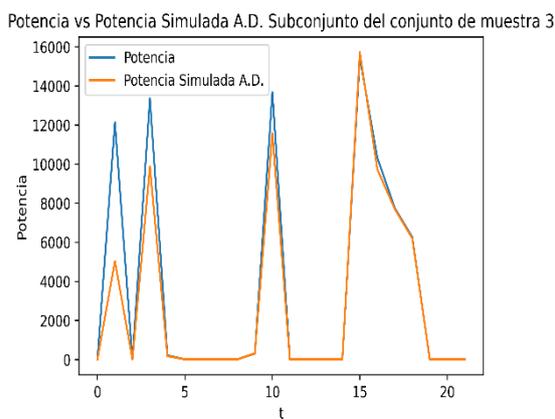


Figura 64: Potencia real frente a potencia estimada por el modelo de árbol de decisión para el subconjunto 3

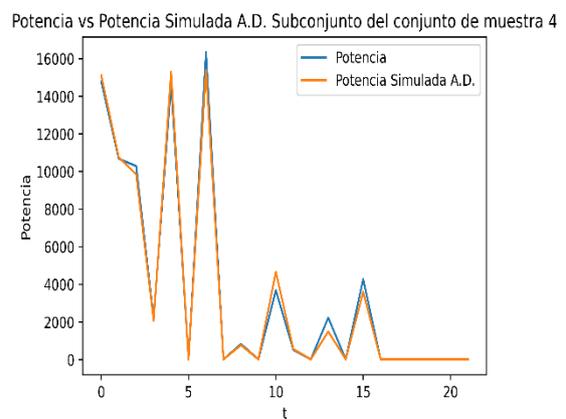


Figura 65: Potencia real frente a potencia estimada por el modelo de árbol de decisión para el subconjunto 4

## 7.7. Observaciones

Como se ha comentado anteriormente, existe un vasto espectro de algoritmos de Machine Learning e Inteligencia Artificial capaces de resolver este tipo de problemas, y en este capítulo sólo se han explorado, de forma superficial, algunos de ellos. Es previsible la obtención de modelos más precisos, haciendo uso de algoritmos más sofisticados, acompañados de una meticulosa ingeniería de características y un refinado ajuste de los hiperparámetros.

Aun así, se dispone en este de modelos con suficiente precisión como para ser utilizados con diversos fines. Entre ellos, se encuentra la posibilidad de utilizar los modelos con el fin de detectar fallos operativos en plantas de generación fotovoltaica.

Estableciendo un margen de tolerancia, tal como cierta desviación durante un periodo 'x', es posible detectar desviaciones significativas que pueden indicar un problema de rendimiento. Por ejemplo, si la producción real es consistentemente menor que la esperada, más allá del margen de tolerancia, esto podría sugerir una disfunción en la planta que necesite ser investigada. De esta manera, los modelos pueden jugar un papel clave en el mantenimiento y optimización de este tipo de instalaciones.

## 8. Conclusiones

En la primera fase de este proyecto, se ha comprobado que la vivienda objeto de estudio cumple con los requisitos del DB-HE1. Esto es de suma importancia debido a la urgente necesidad de controlar y limitar la demanda energética, tanto por su impacto ambiental como por su influencia en el precio de la energía y sus consecuencias asociadas. Es esencial garantizar desde el principio que se cumplan estándares mínimos, especialmente cuando se trata de recursos con una vida útil tan prolongada como una vivienda.

Posteriormente, al modelar la vivienda con DesignBuilder, se ha obtenido una comprensión mucho más precisa de su comportamiento energético. Disponer de un modelo como este nos ha permitido evaluar los efectos de las mejoras tanto antes de la construcción como después. Dada la incertidumbre en factores como el clima a lo largo de la vida útil del edificio, cambios en el uso de la vivienda y modificaciones normativas, contar con un modelo desde el principio resulta invaluable. Además, la gran cantidad de información recopilada puede utilizarse para sensibilizar a los habitantes sobre el impacto de sus elecciones en el uso de los diferentes elementos de la vivienda.

En una etapa posterior, se ha diseñado un programa en Python con diferentes módulos. En primer lugar, los módulos relacionados con el dimensionado de la instalación fotovoltaica permiten hacer un dimensionado mucho más preciso que otros programas o métodos. Esto es debido a la gran cantidad de información con la que se trabaja. Recordando lo expuesto en capítulos anteriores, se trabaja con bases de datos meteorológicas que contienen datos de radiación y temperatura y que, pueden ser extraídos de diversas fuentes, con datos de consumo; que pueden ser extraídos tanto de programas de simulación como de aparatos de medición de consumo eléctrico, y con amplias bases de datos técnicos de paneles fotovoltaicos e inversores. Esta gran cantidad de información junto con su adecuada utilización permite explorar una gran cantidad de combinaciones y obtener, según los criterios establecidos, una combinación de elementos capaces de generar energía a un precio mucho menor que el de la red, y de una fuente renovable. En concreto, para la vivienda analizada, la combinación “ganadora” permite obtener energía eléctrica a aproximadamente dos céntimos por kilovatio-hora.

Por otra parte, el programa permite simular estrategias interesantes como la producción de agua caliente sanitaria en los momentos de excedente energético de la instalación fotovoltaica. Si bien el ahorro monetario obtenido no es significativo en el caso analizado, los conceptos trabajados y los resultados obtenidos resultan muy sugerentes. Según se muestra en el apartado correspondiente, incluso en situaciones poco favorables dada la distribución de los consumos de ACS y de los excedentes, del tamaño del depósito y de aproximaciones hechas durante el cálculo, que incrementan ligeramente los resultados de pérdidas energéticas, se obtiene un ahorro monetario. Es por tanto, una opción sumamente atractiva para analizar, especialmente en escenarios donde no es factible o conveniente inyectar excedentes en la red eléctrica, y en situaciones en las que se cuenta con un depósito de mayor capacidad. Como se ha mencionado previamente, un depósito más grande conlleva menores pérdidas relativas de energía

Otro de los módulos permite simular el escenario resultante de la formación de una comunidad energética. Las comunidades energéticas destacan por su gran interés, ya que ofrecen la posibilidad de redirigir el excedente de energía hacia viviendas locales que la necesiten, lo que a su vez reduce

significativamente las pérdidas de distribución debido a la proximidad. Además, fomentan la cooperación entre vecinos en términos energéticos, lo que conlleva beneficios adicionales, como una mayor conciencia y participación en la promoción de un mundo más sostenible. Los resultados, sin embargo, subrayan la necesidad de un análisis exhaustivo antes de llevar a cabo cualquier idea, por muy atractiva que pueda parecer en principio. En este proyecto, hemos examinado una única configuración de comunidad energética que involucra tres viviendas, y los resultados claramente indican la importancia de explorar alternativas. Estas alternativas podrían incluir la reevaluación de la configuración de la comunidad energética, considerando la incorporación de nuevos miembros y, por lo tanto, ajustando la distribución de consumos para aprovechar de manera más eficiente los excedentes, o incluso la posibilidad de inyectar esos excedentes en la red eléctrica. A pesar de que no se hayan analizado otras configuraciones de comunidades energéticas durante este proyecto, es relevante destacar que el programa desarrollado ofrece la capacidad de realizar múltiples simulaciones y evaluar los resultados según sea necesario. Además, se ha llevado a cabo un análisis de la compensación económica de inyectar los excedentes en la red eléctrica, lo que resulta en un descuento adicional de aproximadamente 106 € y un retorno de la inversión en tan solo un año y tres meses. Esto brinda una opción que, con un desembolso inicial asequible y sostenible, conlleva una notable reducción de la factura eléctrica.

Por último, es importante destacar que nuestro programa cuenta con múltiples módulos y funciones que pueden utilizarse tanto de manera conjunta como independiente, según el propósito específico. Además, este programa es adaptable y ampliable, lo que significa que con el tiempo se puede mejorar su utilidad y precisión, convirtiéndolo en una herramienta cada vez más valiosa.

En la última etapa del proyecto, se ha analizado la aplicación de algunas técnicas de Machine Learning para la previsión de la producción de energía en una planta fotovoltaica. Incluso utilizando algoritmos muy sencillos se han obtenido muy buenos resultados en la previsión de la producción. Se concluye, por tanto, que la exploración e implementación de este tipo de sistemas puede ser muy útil tanto para la detección temprana de fallos en las instalaciones como para su utilización en la gestión de la demanda energética. Especialmente, si se utilizan en conjunto con herramientas de predicción de la demanda y de sistemas automatizados. En su conjunto, estas soluciones conforman una poderosa y eficaz herramienta para un uso más eficiente y responsable de la energía.

## 9. Bibliografía

ASHRAE (American Society of Heating, Refrigerating and Air-Conditioning Engineers).

Aurea Consulting. "Aprende DesignBuilder." Ecoeficiente.es. 20 de octubre de 2022.

<https://ecoeeficiente.es/curso-online-aprende-designbuilder>.

Cares Renewables. "Is a Roof Top Solar Power Plant Installation Safe?"

<https://www.caresrenewables.com/post/is-a-roof-top-solar-power-plant-installation-safe>.

Accedido el 22 de marzo de 2023.

Çengel, Yunus A. *Transferencia de Calor y Masa*. McGRAW-HILL/INTERAMERICANA EDITORES, S.A. DE C.V., 2007.

DesignBuilder Software. "Overview of the Model Data Tabs and Data Inheritance." YouTube. 22 de junio de 2013. <https://www.youtube.com/watch?v=BuzViH7iaXI>.

DesignBuilder Software. "Zoning Blocks by Drawing Partitions, Zone Types and Zoning Protocols." YouTube. 22 de junio de 2013. <https://www.youtube.com/watch?v=BuzViH7iaXI>.

DesignBuilder Software Ltd. DesignBuilder. Versión 7.0.2.006. 2022. <https://designbuilder.co.uk/>.

DesignBuilder Software Ltd. DesignBuilder 2.1 User's Manual. 2009.

[http://www.designbuildersoftware.com/docs/designbuilder/DesignBuilder\\_2.1\\_Users-Manual\\_Ltr.pdf](http://www.designbuildersoftware.com/docs/designbuilder/DesignBuilder_2.1_Users-Manual_Ltr.pdf).

European Commission, Joint Research Centre. Photovoltaic Geographical Information System (PVGIS). [https://re.jrc.ec.europa.eu/pvg\\_tools/en/tools.html](https://re.jrc.ec.europa.eu/pvg_tools/en/tools.html).

Géron, Aurélien. *Aprende Machine Learning con Scikit-Learn, Keras y TensorFlow*. Madrid: Anaya Multimedia, 2020. Originalmente publicado como *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. Sebastopol, CA: O'Reilly, 2019.

Harris, Charles R., et al. "Array programming with NumPy." *Nature* 585, no. 7825 (2020): 357-362, <https://numpy.org/doc/stable/reference/index.html>.

IDAE (Instituto para la Diversificación y Ahorro de la Energía). *Consumos del Sector Residencial en España. Resumen de Información Básica. Apartado 8: Consumo Medio Anual de Electrodomésticos por Hogar*.

IDAE (Instituto para la Diversificación y Ahorro de la Energía). *Guía Técnica: Diseño y cálculo del aislamiento térmico de conducciones, aparatos y equipos*.

IDAE (Instituto para la Diversificación y Ahorro de la Energía). *Pliego de Condiciones Técnicas de Instalaciones Conectadas a Red*.

IDAE (Instituto para la Diversificación y Ahorro de la Energía). *SPAHOUSEC II: Análisis estadístico del consumo de gas natural en las viviendas principales con calefacción individual. Tabla 5.8. Consumo medio (kWh/hogar) de gas natural por usos según zonas climáticas y tipo de viviendas*.

Incropera, Frank P., y David P. DeWitt. *Fundamentos de Transferencia de Calor*. Pearson Educación, 1999.

López Landa, L. "Estudio comparativo de las diferentes soluciones de instalación térmica para una vivienda unifamiliar." Trabajo de fin de grado, Universidad de Cantabria, 2017. Director: Carcedo Haya, J.

MathWorks. "MPPT Algorithm." <https://www.mathworks.com/solutions/electrification/mppt-algorithm.html>. Accedido el 25 de marzo de 2023.

Ministerio de Industria, Comercio y Turismo. Reglamento Electrotécnico de Baja Tensión.

Ministerio de Transportes, Movilidad y Agenda Urbana. "Código Técnico de la Edificación. Documento Básico de Ahorro de Energía." 2019.

Ministerio de Transportes, Movilidad y Agenda Urbana. "Código Técnico de la Edificación. Documento Básico de Ahorro de Energía (DB-HE). Anexo B." 2019.

Ministerio de Transportes, Movilidad y Agenda Urbana. "Código Técnico de la Edificación. Documento Básico de Ahorro de Energía (DB-HE). Anexo F." 2019.

Ministerio de Transportes, Movilidad y Agenda Urbana. "Código Técnico de la Edificación. Documento Básico de Ahorro de Energía (DBHE). Anexo D. Tabla A." 2019.

Ministerio de Transportes, Movilidad y Agenda Urbana. "Guía de aplicación DB HE 2019 - Ejemplos (I). Código Técnico de la Edificación. B. Arquitectura y definición geométrica, 2. Planos: Situación y orientación. Plantas. Secciones. Alzados." 2019.

Ministerio de Transportes, Movilidad y Agenda Urbana. "Guía de aplicación DB HE 2019 - Ejemplos (I). Código Técnico de la Edificación. C. Definición constructiva, 1. Composición de los cerramientos. Opacos y huecos." 2019.

Ministerio de Transportes, Movilidad y Agenda Urbana. Guía IDAE 021: Guía Profesional de Tramitación del Autoconsumo (edición v.5.1). Madrid, enero de 2023. Autor: Departamento Solar y Autoconsumo del IDAE. Grupo de Trabajo de Autoconsumo de ENERAGEN.

pandas development team. Pandas: Python Data Analysis Library, n.d., <https://pandas.pydata.org>.

Parlamento Europeo y Consejo. "Directiva 2009/28/CE del Parlamento Europeo y del Consejo, de 23 de abril de 2009, relativa al fomento del uso de energía procedente de fuentes renovables y por la que se modifican y se derogan las Directivas 2001/77/CE y 2003/30/CE." 2009.

Pedregosa, F. et al. "Scikit-learn: Machine Learning in Python." Journal of Machine Learning Research 12 (2011): 2825-2830, [https://scikit-learn.org/stable/user\\_guide.html](https://scikit-learn.org/stable/user_guide.html).

Real Decreto 244/2019, de 5 de abril. Condiciones administrativas, técnicas y económicas del autoconsumo de energía eléctrica. Diario Oficial de la Unión Europea, 2019.

SMA Solar Technology AG. "Ficha Técnica del Inversor Sunny Boy 2.5 KW." <https://www.sma.de/en/products/solarinverters/sunny-boy-15-20-25#c81>. Accedido el 25 de marzo de 2023.

Samuel, Arthur. "Some Studies in Machine Learning Using the Game of Checkers." IBM Journal of Research and Development, 1959.

UNE-EN. 2017. "12207: Ventanas y Puertas. Permeabilidad al Aire. Clasificación." Madrid, España: Asociación Española de Normalización.

# CERRAMIENTOS

## Cerramientos opacos

MURO EXTERIOR			M EXT
COMPOSICIÓN DEL CERRAMIENTO			SECCIÓN CONSTRUCTIVA
capas	espesor (m)	Cond. $\lambda$ (W/m·K)	
1 Mortero de cemento	0,03	0,550	
2 EPS Poliestireno	0,14	0,038	
3 1 pie LP métrico o catalán	0,24	0,667	
4 Mortero de cemento	0,01	0,550	
5 Cámara de aire sin ventilar vertical	0,05	0,18	
6 Placa de yeso laminado	0,015	0,250	
7 Placa de yeso laminado	0,015	0,250	
<b>Total</b>	<b>0,500</b>		
<b>TRANSMITANCIA</b>	<b>0,22 W/m<sup>2</sup>K</b>		

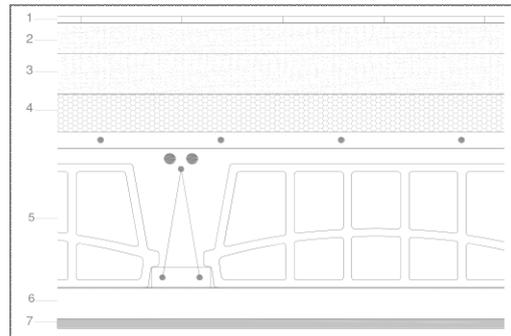
MURO DE CÁMARA SANITARIA			MURO CAM SANIT
COMPOSICIÓN DEL CERRAMIENTO			SECCIÓN CONSTRUCTIVA
capas	espesor (m)	Cond. $\lambda$ (W/m·K)	
1 Betún fieltro o lámina	0,009	0,23	
2 1 pie LP métrico o catalán	0,24	0,667	
<b>Total</b>	<b>0,249</b>		
<b>TRANSMITANCIA*</b>	<b>1,76 W/M2k</b>		

\* Sin corregir por contacto

TABIQUE INTERIOR			TAB INT
COMPOSICIÓN DEL CERRAMIENTO			SECCIÓN CONSTRUCTIVA
capas	espesor (m)	Cond. $\lambda$ (W/m·K)	
1 Placa de yeso laminado	0,013	0,250	
2 Placa de yeso laminado	0,013	0,250	
3 MW Lana mineral	0,04	0,041	
4 Placa de yeso laminado	0,013	0,250	
5 Placa de yeso laminado	0,013	0,250	
<b>Total</b>	<b>0,092</b>		
<b>TRANSMITANCIA*</b>	<b>0,46 W/m<sup>2</sup>K</b>		

**FORJADO INTERIOR ENTRE PLANTAS (ACOND-ACOND)**
**FOR INT**
**COMPOSICIÓN DEL CERRAMIENTO**

capas	espesor (m)	Cond. $\lambda$ (W/m·K)
1 Plaqueta de gres	0,01	2,300
2 Mortero de cemento	0,05	0,550
3 Mortero de cemento de difusión	0,065	0,550
4 XPS Expandido	0,06	0,034
5 FU entrevigado cerámico	0,25	0,908
6 Cámara de aire sin ventilar horizontal	0,05	0,160
7 Placa de yeso laminado	0,015	0,250
<b>Total</b>	<b>0,500</b>	

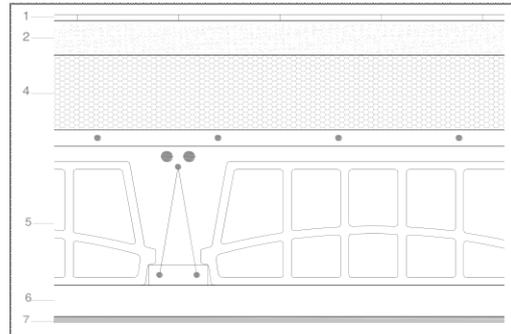
**TRANSMITANCIA\***      **0,37 W/m<sup>2</sup>K**
**SECCIÓN CONSTRUCTIVA**


\* Sin corregir por contacto

**FORJADO INTERIOR ENTRE PLANTAS (ACOND-NO HAB) \***
**FOR INT AC-NH**
*Forjado entre espacio acondicionado y no habitable (opción 1 y parcialmente opción 2)*
**COMPOSICIÓN DEL CERRAMIENTO**

capas	espesor (m)	Cond. $\lambda$ (W/m·K)
1 Plaqueta de gres	0,01	2,300
2 Mortero de cemento	0,055	0,550
3 XPS Expandido ( $\lambda:0,034$ W/m·K)	0,12	0,034
4 FU entrevigado cerámico	0,25	0,908
5 Cámara de aire sin ventilar horizontal	0,05	0,160
6 Placa de yeso laminado	0,015	0,250
<b>Total</b>	<b>0,500</b>	

**TRANSMITANCIA**      \*      **0,16 W/m<sup>2</sup>K**

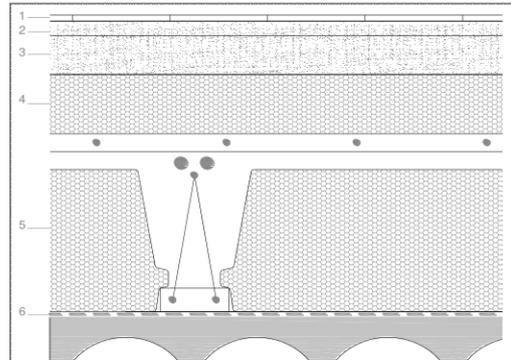
    \*\*      **0,17 W/m<sup>2</sup>K**
**SECCIÓN CONSTRUCTIVA**


\* Corregida por contacto con N.H. en toda la planta

\*\* Corregida por contacto parcial con N.H. en una parte de la planta

**FORJADO CON CÁMARA SANITARIA**
**FOR CAM SANIT**
**COMPOSICIÓN DEL CERRAMIENTO**

capas	espesor (m)	Cond. $\lambda$ (W/m·K)
1 Plaqueta de gres	0,01	2,300
2 Mortero de cemento	0,025	0,550
3 Mortero de cemento de difusión	0,06	0,550
4 XPS Expandido	0,1	0,034
5 Forjado Entrevigado EPS mecanizado	0,3	0,256
6 Lámina de cloruro de polivinilo (PVC)	0,005	0,170
<b>Total</b>	<b>0,500</b>	

**TRANSMITANCIA\***      **0,21 W/m<sup>2</sup>K**
**SECCIÓN CONSTRUCTIVA**


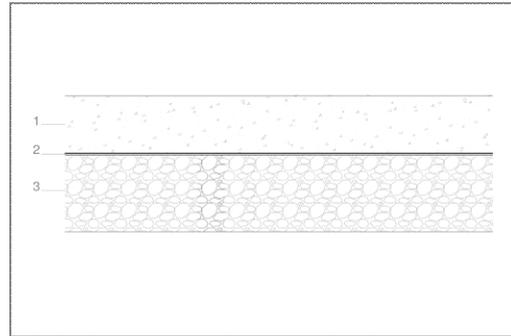
**SOLERA DE CÁMARA SANITARIA**

SOL CAM SANIT

COMPOSICIÓN DEL CERRAMIENTO

capas	espesor (m)	Cond. $\lambda$ (W/m·K)
1 Hormigón en masa	0,08	1,650
2 Lámina PVC	0,003	0,170
3 Arena y grava	0,12	2,000
<b>Total</b>	<b>0,203</b>	

SECCIÓN CONSTRUCTIVA



TRANSMITANCIA 2,98 W/m<sup>2</sup>K

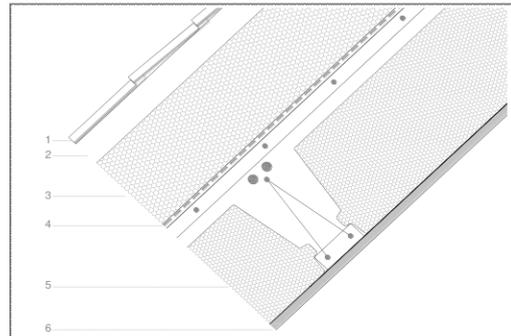
**CUBIERTA INCLINADA DE TEJA**

CUB IN TEJA

COMPOSICIÓN DEL CERRAMIENTO

capas	espesor (m)	Cond. $\lambda$ (W/m·K)
1 Teja de arcilla cocida	0,015	1,000
2 Cámara aire ventilada, flujo ascend.	0,05	0,060
3 XPS Expandido	0,16	0,034
4 Betún fieltro o lámina	0,01	0,230
5 Forjado Entrevigado EPS mecanizado	0,25	0,266
6 Placa de yeso laminado	0,015	0,250
<b>Total</b>	<b>0,500</b>	

SECCIÓN CONSTRUCTIVA



TRANSMITANCIA 0,17 W/m<sup>2</sup>K

Huecos

**CONFIGURACIÓN HUECOS EN FACHADA NORTE Y ESTE**

N/E

La posición relativa de las carpinterías en el muro es alineada a su cara interior

**VIDRIO**

Tipo de vidrio	Doble bajo emisivo < 0,03
Composición	4-16 Argón-4
Factor solar	0,61
Transmitancia térmica	1,0 W/m <sup>2</sup> K

**MARCO**

Material	PVC
Rotura pte. térmico	-
Color carpintería	Blanco claro
Absortividad	0,3
Trasmirancia térmica	1,5 W/m <sup>2</sup> K
Porcentaje marco	25 % del hueco

Protecciones móviles

Elemento	Persianas exteriores
Color	Blanco medio
ggl;sh,wi	0,05

Permeabilidad al aire	9 m <sup>3</sup> /h·m <sup>2</sup>
	Clase 3
<b>U<sub>HUECO</sub></b>	<b>1,14 W/m<sup>2</sup>·K</b>

**CONFIGURACIÓN HUECOS EN FACHADA SUR Y OESTE**

S/O

La posición relativa de las carpinterías en el muro es alineada a su cara interior

**VIDRIO****MARCO**

Tipo de vidrio	Doble bajo emisivo < 0,03
Composición	4-16 Argón-4
Factor solar	0,42
Transmitancia térmica	1,0 W/m <sup>2</sup> K

Material	PVC
Rotura pte. térmico	-
Color carpintería	Blanco claro
Absortividad	0,3
Trasmirancia térmica	1,5 W/m <sup>2</sup> K
Porcentaje marco	25 % del hueco

Protecciones móviles

Elemento	Persianas exteriores
Color	Blanco medio
ggl;sh,wi	0,05

Permeabilidad al aire	9 m <sup>3</sup> /h·m <sup>2</sup>
	Clase 3
<b>U<sub>HUECO</sub></b>	1,14 W/m <sup>2</sup> ·K

**CONFIGURACIÓN HUECO PUERTA ACCESO**

PTA

La posición relativa de las carpinterías en el muro es alineada a su cara interior

**VIDRIO****MARCO**

Tipo de vidrio	-
Composición	-
Factor solar	-
Transmitancia térmica	- W/m <sup>2</sup> K

Material	Madera densidad media alta
Rotura pte. térmico	-
Color carpintería	Blanco claro
Absortividad	0,3
Trasmirancia térmica	2,2 W/m <sup>2</sup> K
Porcentaje marco	100 % del hueco

Protecciones móviles

Elemento	Persianas exteriores
Color	Blanco medio
ggl;sh,wi	0,05

Permeabilidad al aire	60 m <sup>3</sup> /h·m <sup>2</sup>
<b>U<sub>HUECO</sub></b>	2,2 W/m <sup>2</sup> ·K

**CONFIGURACIÓN LUCERNARIOS SUR Y ESTE**

S/E

La posición relativa de las carpinterías en el muro es alineada a su cara exterior

**VIDRIO****MARCO**

Tipo de vidrio	Doble bajo emisivo < 0,03
Composición	4-16 Argón-4
Factor solar	0,42
Transmitancia térmica	1,0 W/m <sup>2</sup> K

Material	Madera densidad media alta
Rotura pte. térmico	-
Color carpintería	Blanco claro
Absortividad	0,3
Trasmirancia térmica	2,4 W/m <sup>2</sup> K
Porcentaje marco	20 % del hueco

Protecciones móviles

Elemento	Persianas exteriores
Color	Blanco medio
ggl;sh,wi	0,05

Permeabilidad al aire	9 m <sup>3</sup> /h·m <sup>2</sup>
	Clase 3
<b>U<sub>HUECO</sub></b>	1,29 W/m <sup>2</sup> ·K

**CONFIGURACIÓN LUCERNARIOS NORTE**

N

La posición relativa de las carpinterías en el muro es alineada a su cara exterior

**VIDRIO**

Tipo de vidrio	Doble bajo emisivo < 0,03
Composición	4-16 Argón-4
Factor solar	0,61
Transmitancia térmica	1,0 W/m <sup>2</sup> K

Protecciones móviles

Elemento	Persianas exteriores
Color	Blanco medio
ggl;sh,wi	0,05

**MARCO**

Material	Madera densidad media alta
Rotura pte. térmico	-
Color carpintería	Blanco claro
Absortividad	0,3
Trasmirancia térmica	2,4 W/m <sup>2</sup> K
Porcentaje marco	20 % del hueco

Permeabilidad al aire	9 m <sup>3</sup> /h·m <sup>2</sup> <i>Clase 3</i>
<b>U<sub>HUECO</sub></b>	<b>1,29 W/m<sup>2</sup>·K</b>

# ANEXOS

# ANEXO 1. Relación del trabajo con los objetivos de desarrollo sostenible de la agenda 2030



**BOUPV**

Butlletí Oficial de la Universitat Politècnica de València

Núm. 118/2022

27/07/2022

16

## ANEXO I. RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030

### Anexo al Trabajo de Fin de Grado y Trabajo de Fin de Máster: Relación del trabajo con los Objetivos de Desarrollo Sostenible de la agenda 2030

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.		●		
ODS 2. Hambre cero.				●
ODS 3. Salud y bienestar.		●		
ODS 4. Educación de calidad.				●
ODS 5. Igualdad de género.				●
ODS 6. Agua limpia y saneamiento.				●
ODS 7. Energía asequible y no contaminante.	●			
ODS 8. Trabajo decente y crecimiento económico.				●
ODS 9. Industria, innovación e infraestructuras.			●	
ODS 10. Reducción de las desigualdades.			●	
ODS 11. Ciudades y comunidades sostenibles.	●			
ODS 12. Producción y consumo responsables.	●			
ODS 13. Acción por el clima.	●			
ODS 14. Vida submarina.			●	
ODS 15. Vida de ecosistemas terrestres.			●	
ODS 16. Paz, justicia e instituciones sólidas.				●
ODS 17. Alianzas para lograr objetivos.	●			

Descripción de la alineación del TFG/TFM con los ODS con un grado de relación más alto.

\*\*\*Utilice tantas páginas como sea necesario.

Editor: Secretaría General / UPV · D.L.: V-5092-2006 · ISSN: 1887-2298 · www.upv.es/secgen

Firmat digitalment per  
UNIVERSITAT POLITÈCNICA DE VALÈNCIA  
Data: 27/07/2022 18:54:40 CEST

Id: UPV-GENFima-128544

Cod. Verificació: Q0W97FCA0C7FRQ01

En este proyecto, se han abordado varios Objetivos de Desarrollo Sostenible (ODS) que son fundamentales para promover un enfoque más sostenible en el ámbito de la energía y la construcción. A continuación, se destacará cómo las actividades y resultados de este proyecto se relacionan con los ODS de Energía Asequible y No Contaminante (ODS 7), Producción y Consumo Responsables (ODS 12), Acción por el Clima (ODS 13) y Alianzas para lograr objetivos (ODS 17).

**1. Energía Asequible y No Contaminante (ODS 7):** El proyecto se centra en la eficiencia energética y la producción de energía asequible y sostenible. A través del dimensionado preciso de una instalación fotovoltaica, se ha logrado generar energía eléctrica a un precio significativamente menor que el de la red, promoviendo así la disponibilidad de energía asequible. Además, se ha explorado la producción de agua caliente sanitaria utilizando excedentes energéticos, lo que contribuye a un uso más eficiente de la energía y ahorro de recursos.

**2. Ciudades y Comunidades Sostenibles (ODS 11):** Este proyecto también tiene implicaciones directas para el ODS 11, que se centra en lograr que las ciudades y comunidades sean más sostenibles. Al optimizar la eficiencia energética en el diseño y operación de viviendas y al promover la formación de comunidades energéticas, se contribuye a la creación de entornos urbanos más sostenibles. La gestión eficiente de la energía y el fomento de la colaboración entre vecinos en la producción y consumo de energía contribuyen a reducir la huella ecológica de las comunidades locales, promoviendo así la sostenibilidad urbana. Este proyecto demuestra cómo la innovación y la planificación adecuada pueden hacer que las ciudades y comunidades sean más responsables desde el punto de vista energético y medioambiental, alineando con el ODS 11.

**3. Producción y Consumo Responsables (ODS 12):** Este proyecto promueve la producción y el consumo responsables al optimizar el dimensionado de la instalación fotovoltaica y simular estrategias para el uso eficiente de la energía. Se busca reducir el consumo de recursos y minimizar las pérdidas energéticas, lo que concuerda con el objetivo de consumo responsable. Además, se ha analizado la posibilidad de formar comunidades energéticas, lo que fomenta la cooperación entre vecinos y una mayor conciencia sobre el consumo responsable de energía.

**4. Acción por el Clima (ODS 13):** El proyecto aborda directamente la acción por el clima al promover la generación de energía a partir de fuentes renovables, como la energía fotovoltaica. La reducción de las emisiones de gases de efecto invernadero y la mitigación del cambio climático son objetivos clave en la acción por el clima. Además, al utilizar técnicas de Machine Learning para prever la producción de energía en una planta fotovoltaica, se contribuye a una gestión más eficiente de la energía, lo que puede ayudar a reducir aún más la huella de carbono.

**5. Alianzas para Lograr Objetivos (ODS 17):** Las comunidades energéticas implican la cooperación entre vecinos y partes interesadas locales para compartir y redistribuir la energía de manera eficiente. Estas alianzas promueven la colaboración en la gestión de recursos energéticos y la reducción de pérdidas de distribución debido a la proximidad entre las viviendas. Al fomentar la colaboración y la participación de la comunidad, se contribuye al fortalecimiento de la cooperación y la construcción de alianzas para abordar desafíos energéticos comunes, alineando así este proyecto con el ODS 17.

## ANEXO 2. Figuras

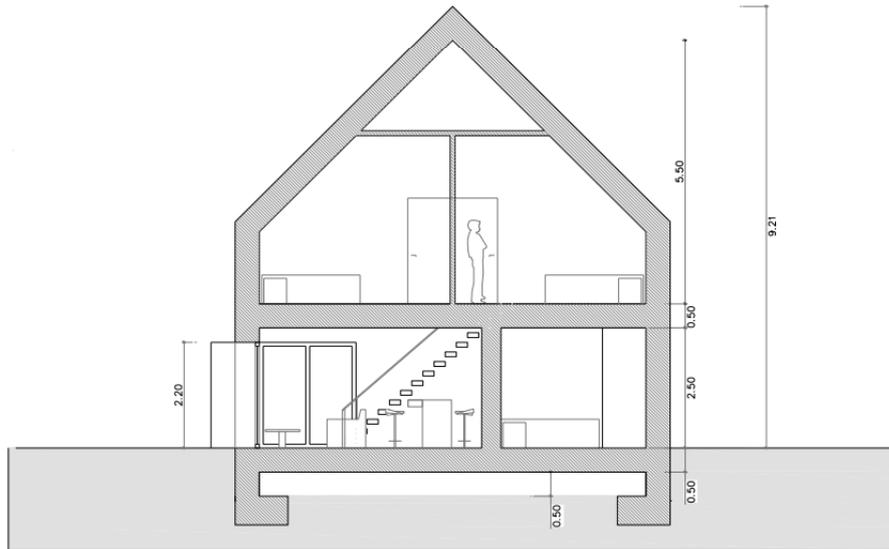


Figura 1: Sección A-A

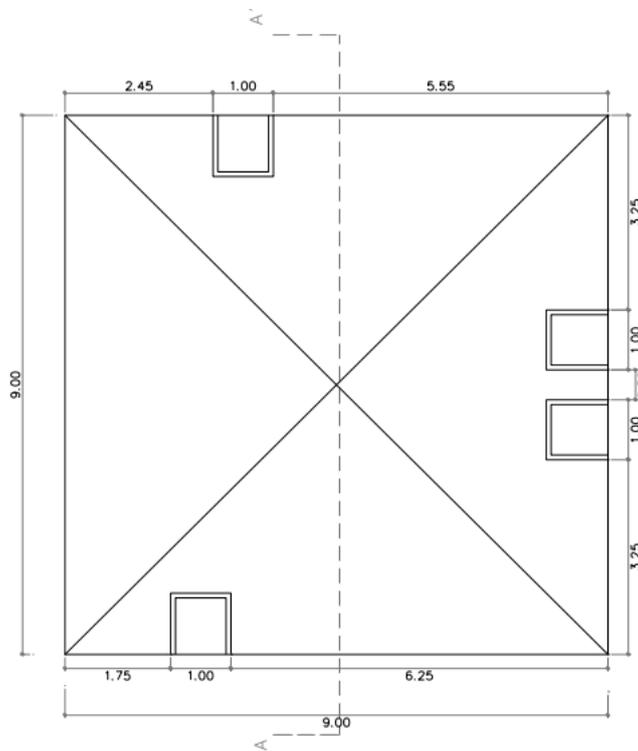


Figura 2: Planta de cubierta

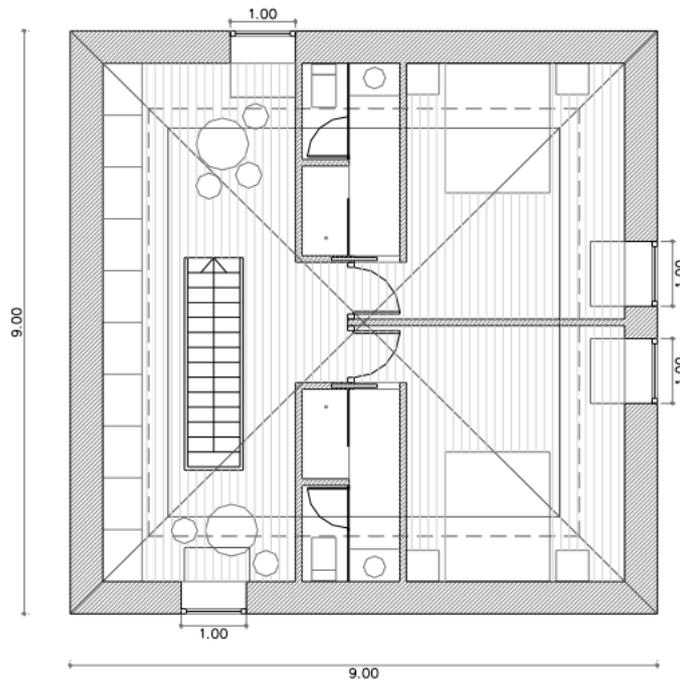


Figura 3: Planta bajocubierta

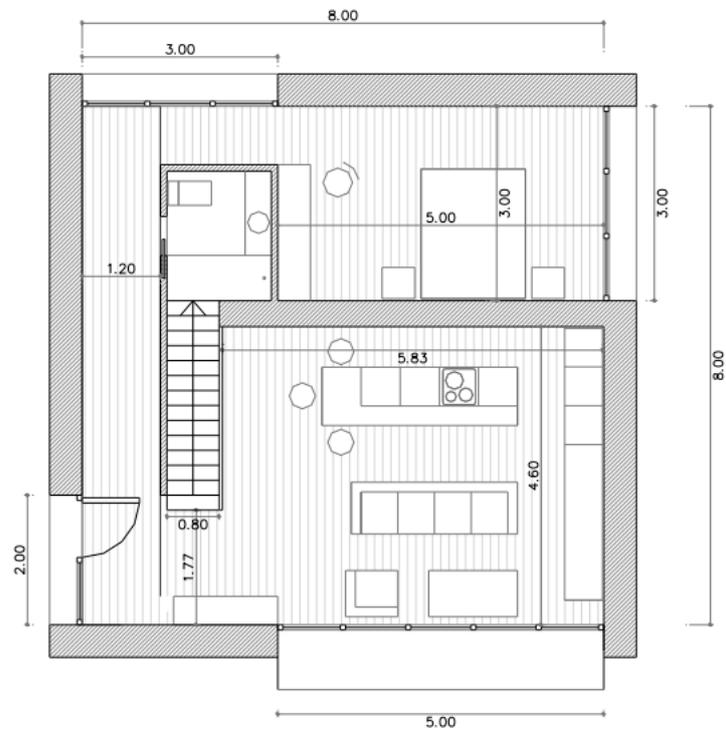


Figura 4: Planta baja

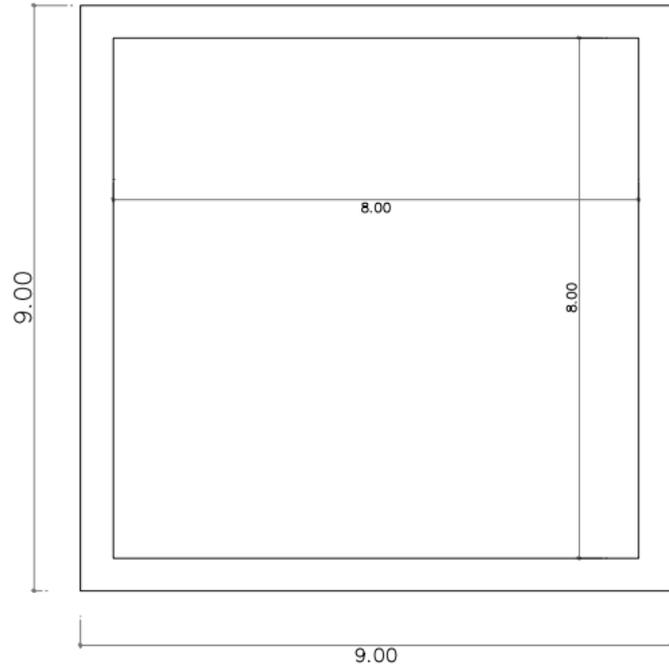


Figura 5: Nivel cámara sanitaria



Figura 6: Alzados

### ANEXO 3. Ganancias solares de las ventanas exteriores

Fecha/Hora	Gan. Solares Ventanas Ext. [kW]
15/07/2002 0:30	0,00000
15/07/2002 1:00	0,00000
15/07/2002 1:30	0,00000
15/07/2002 2:00	0,00000
15/07/2002 2:30	0,00000
15/07/2002 3:00	0,00000
15/07/2002 3:30	0,00000
15/07/2002 4:00	0,00000
15/07/2002 4:30	0,00000
15/07/2002 5:00	0,00000
15/07/2002 5:30	0,00000
15/07/2002 6:00	0,00165
15/07/2002 6:30	0,03950
15/07/2002 7:00	0,08080
15/07/2002 7:30	0,11129
15/07/2002 8:00	0,13559
15/07/2002 8:30	0,15571
15/07/2002 9:00	0,17211
15/07/2002 9:30	0,18448
15/07/2002 10:00	0,19375
15/07/2002 10:30	0,19861
15/07/2002 11:00	0,20195
15/07/2002 11:30	0,20274
15/07/2002 12:00	0,20137
15/07/2002 12:30	0,19839
15/07/2002 13:00	0,19273
15/07/2002 13:30	0,18463
15/07/2002 14:00	0,17445
15/07/2002 14:30	0,16300
15/07/2002 15:00	0,15149
15/07/2002 15:30	0,14046
15/07/2002 16:00	0,12975
15/07/2002 16:30	0,11770
15/07/2002 17:00	0,10524
15/07/2002 17:30	0,09190
15/07/2002 18:00	0,07689
15/07/2002 18:30	0,08560
15/07/2002 19:00	0,06990
15/07/2002 19:30	0,04830
15/07/2002 20:00	0,01680
15/07/2002 20:30	0,00000

15/07/2002 21:00	0,00000
15/07/2002 21:30	0,00000
15/07/2002 22:00	0,00000
15/07/2002 22:30	0,00000
15/07/2002 23:00	0,00000
15/07/2002 23:30	0,00000
<b>Total (Día)</b>	<b>3,83000</b>
<b>Total (Mes)</b>	<b>114,80000</b>

## ANEXO 4. Galería de Imágenes del Modelo de Vivienda en DesignBuilder



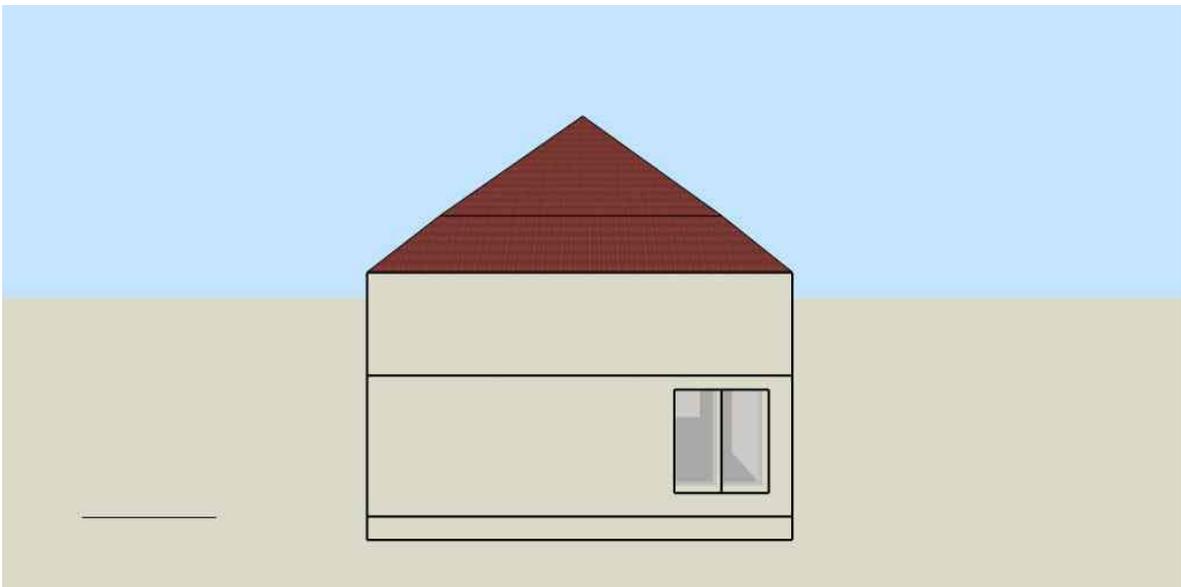
*Anexo 2, Imagen 1: Fachada Norte*



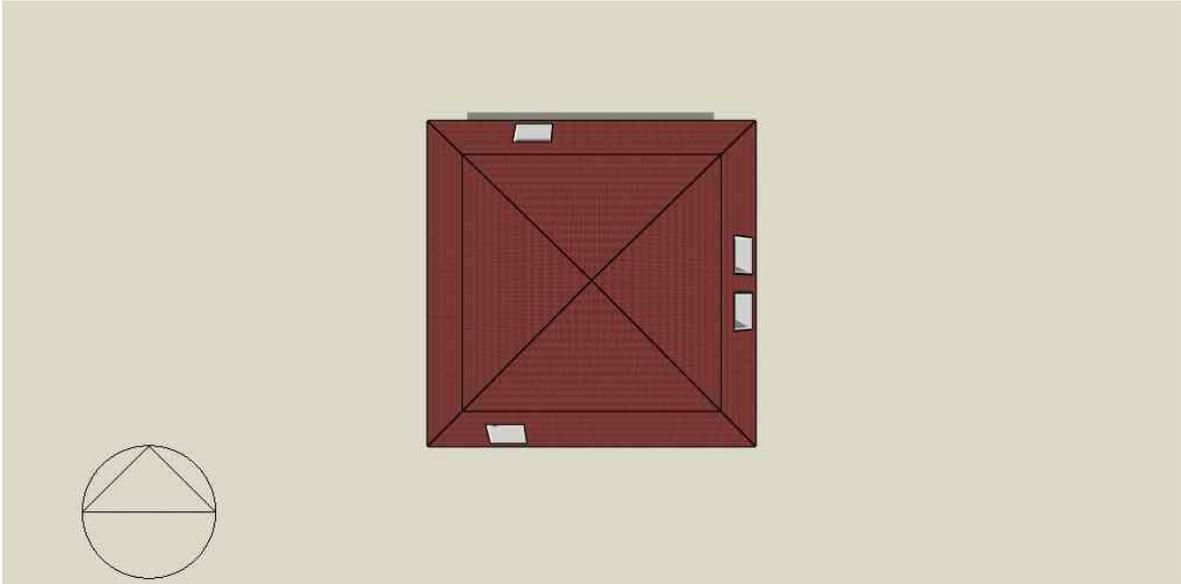
*Anexo 2, Imagen 2: Fachada Sur*



Anexo 2, Imagen 3: Fachada Este



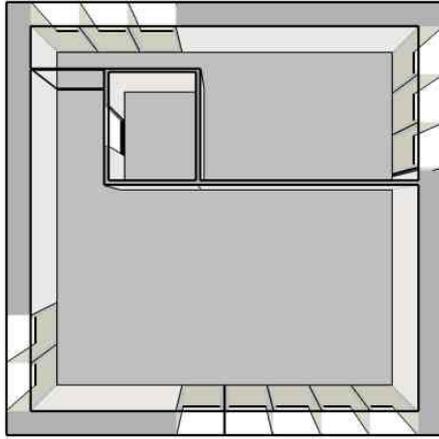
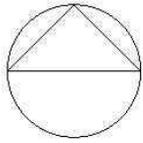
Anexo 2, Imagen 4: Fachada Oeste



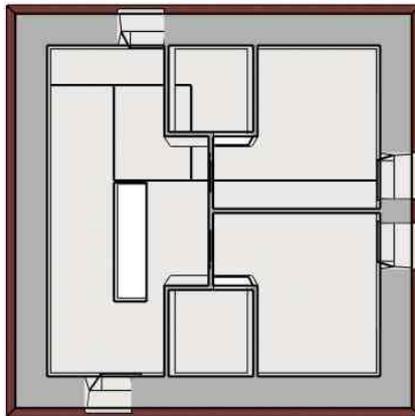
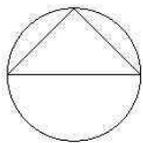
Anexo 2, Imagen 5: Planta



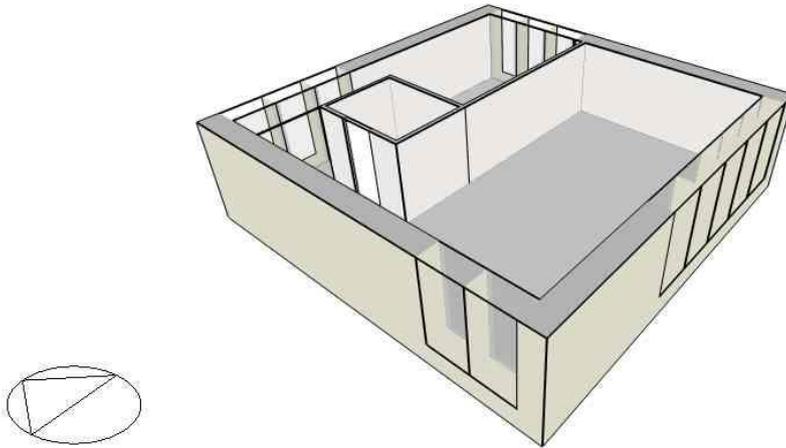
Anexo 2, Imagen 6: Vista axonométrica



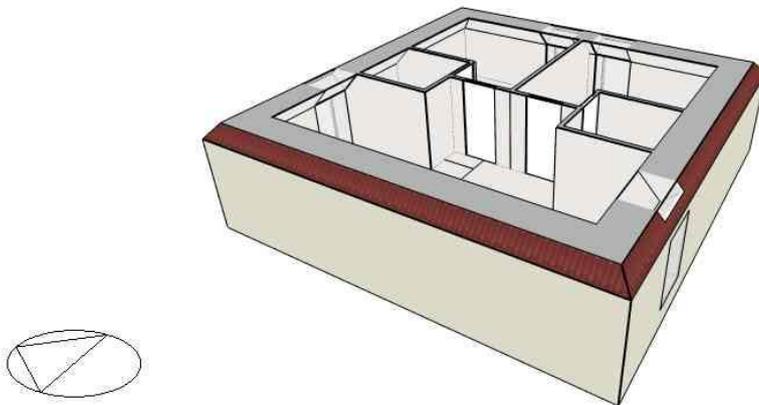
Anexo 2, Imagen 7: Interior 1ª Planta – Planta



Anexo 2, Imagen 8: Interior 2ª Planta – Planta

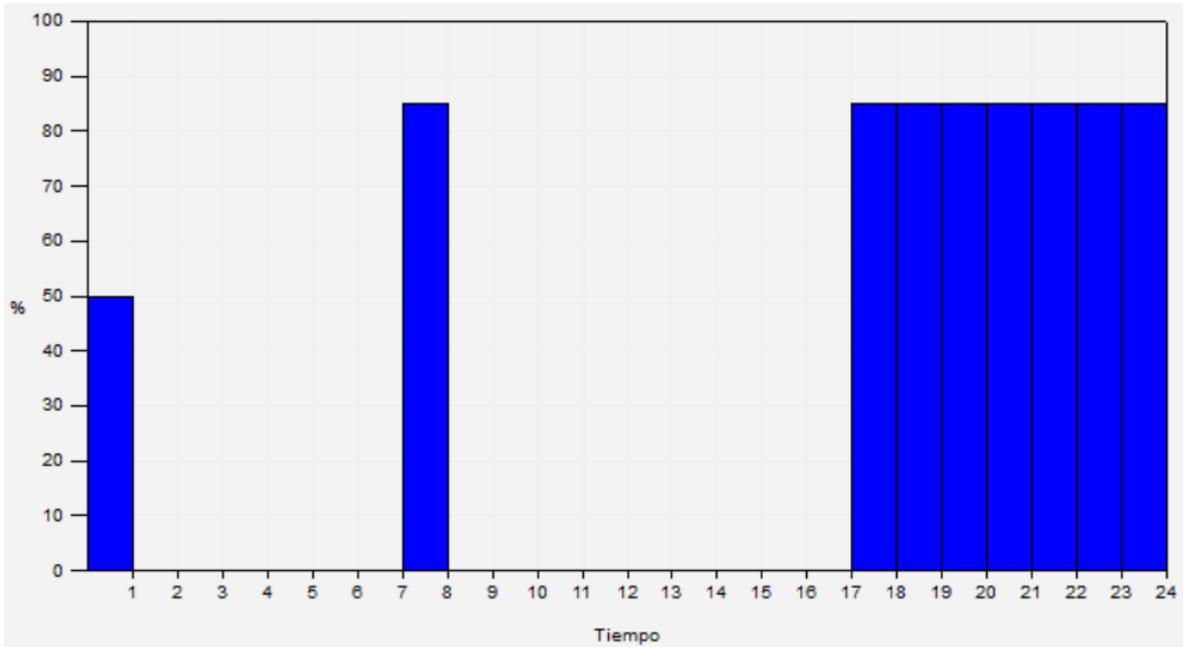


Anexo 2, Imagen 9: Interior 1ª Planta - Vista axonométrica

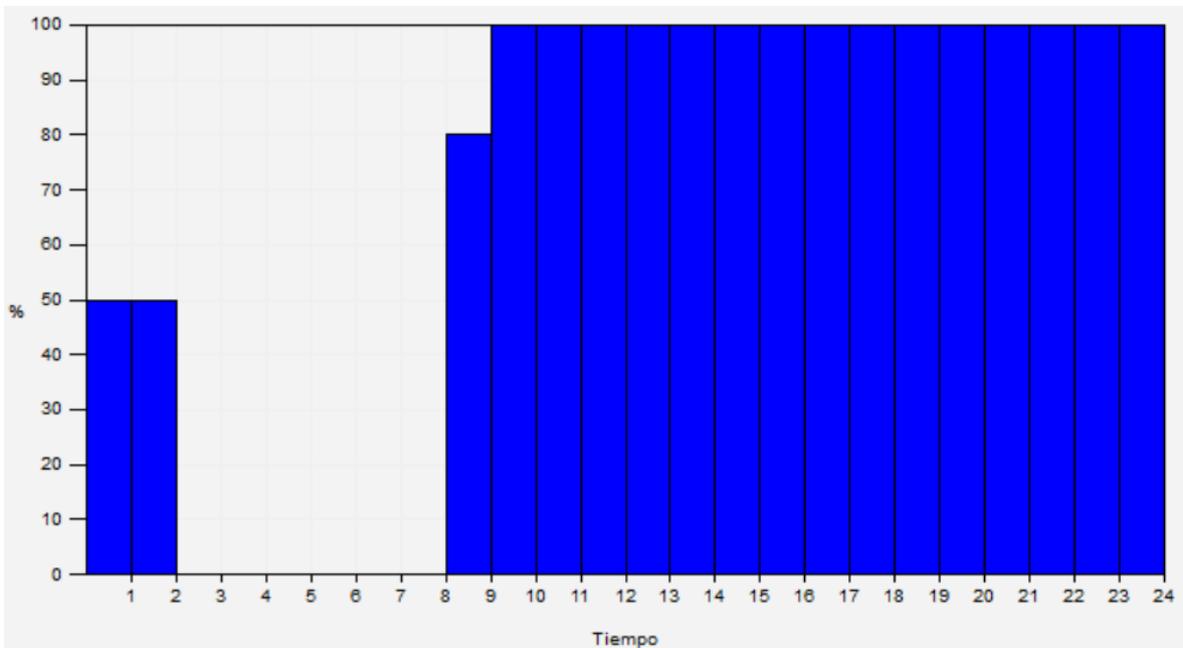


Anexo 2, Imagen 10: Interior 2ª Planta - Vista axonométrica

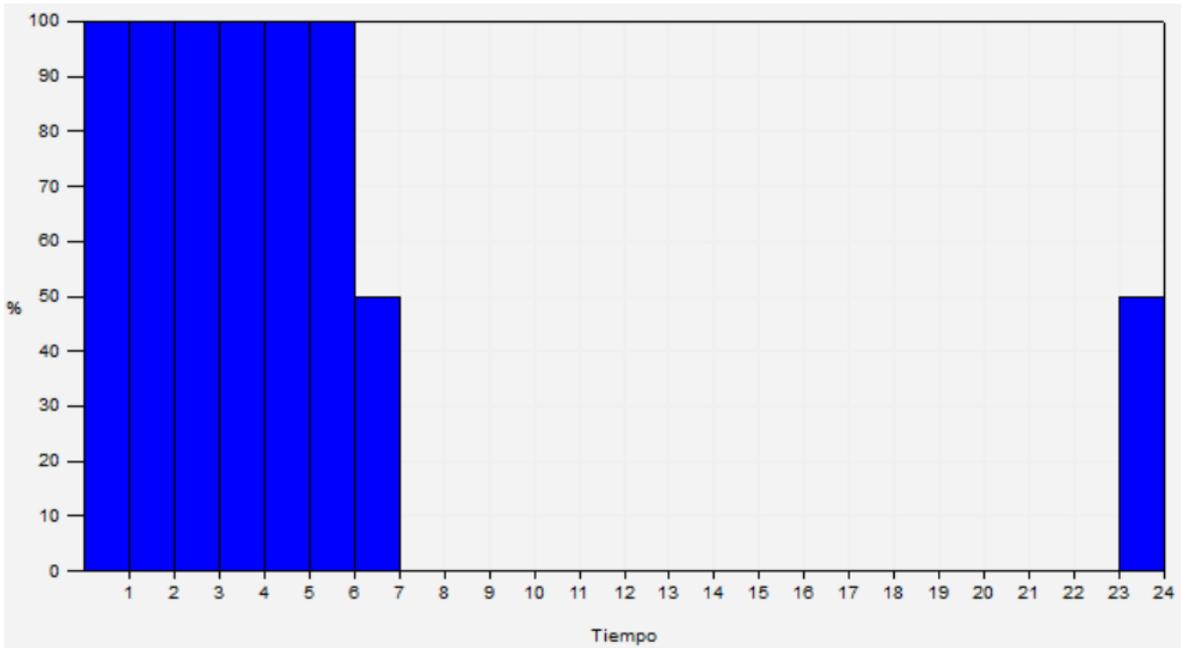
## ANEXO 5: Perfiles de Ocupación



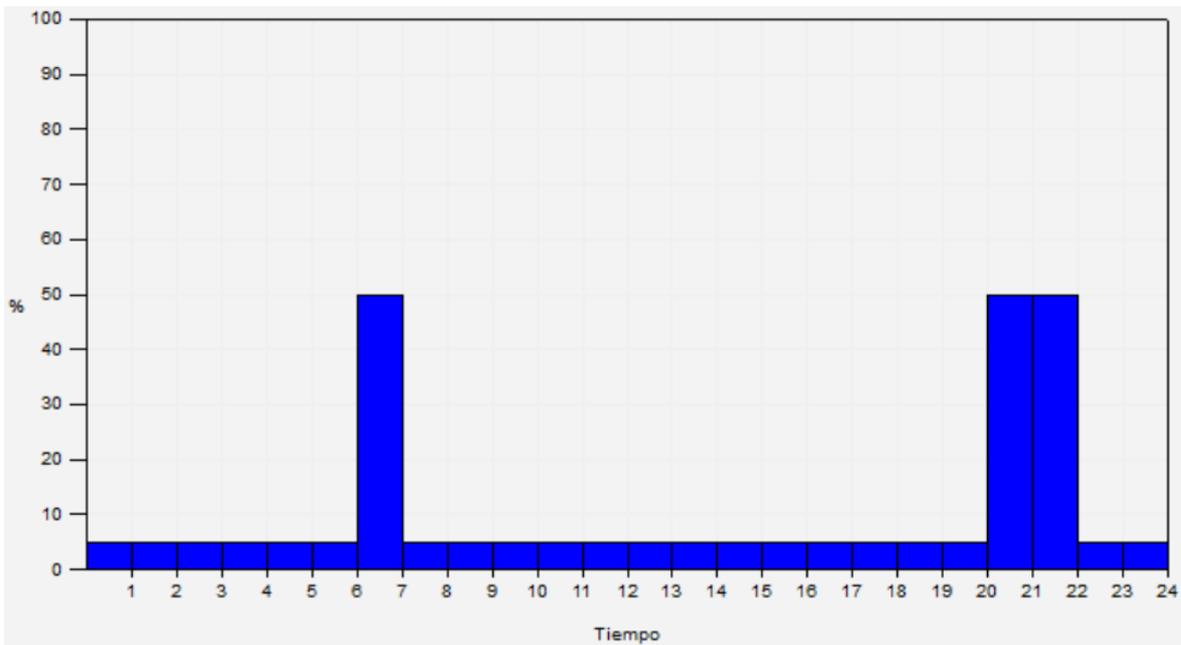
Anexo 3, Imagen 1: Perfil de Ocupación: Zona de Estar y Cocina (Laborable)



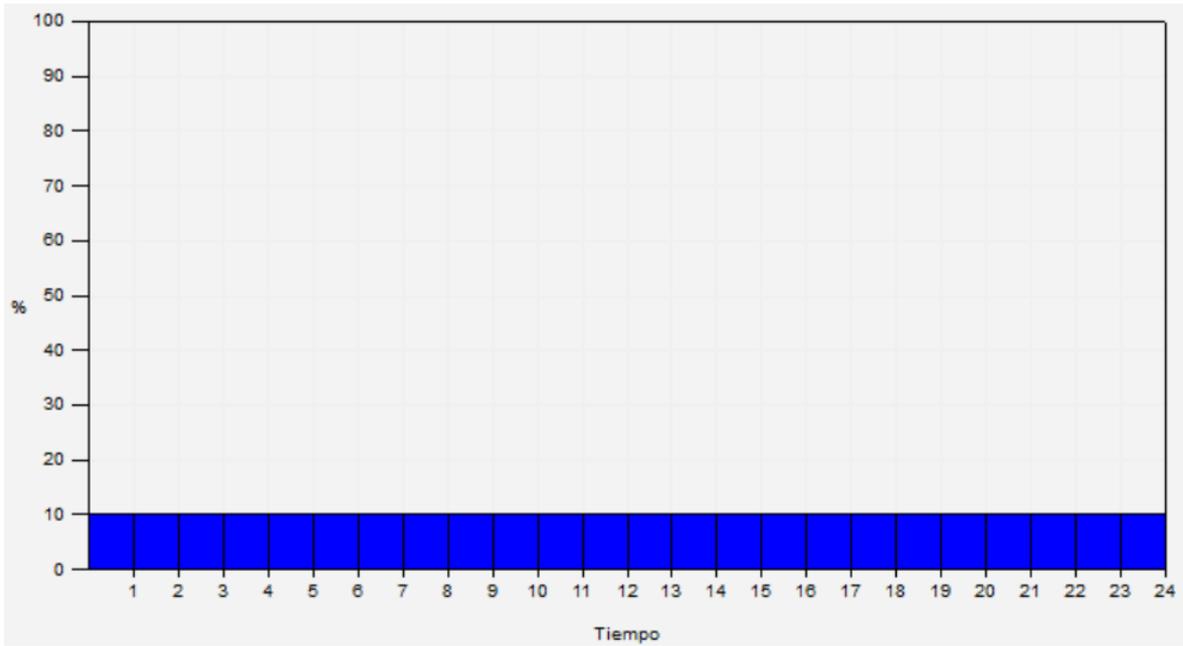
Anexo 3, Imagen 2: Perfil de Ocupación: Zona de Estar y Cocina (Festivo)



Anexo 3, Imagen 3: Perfil de Ocupación: Dormitorios

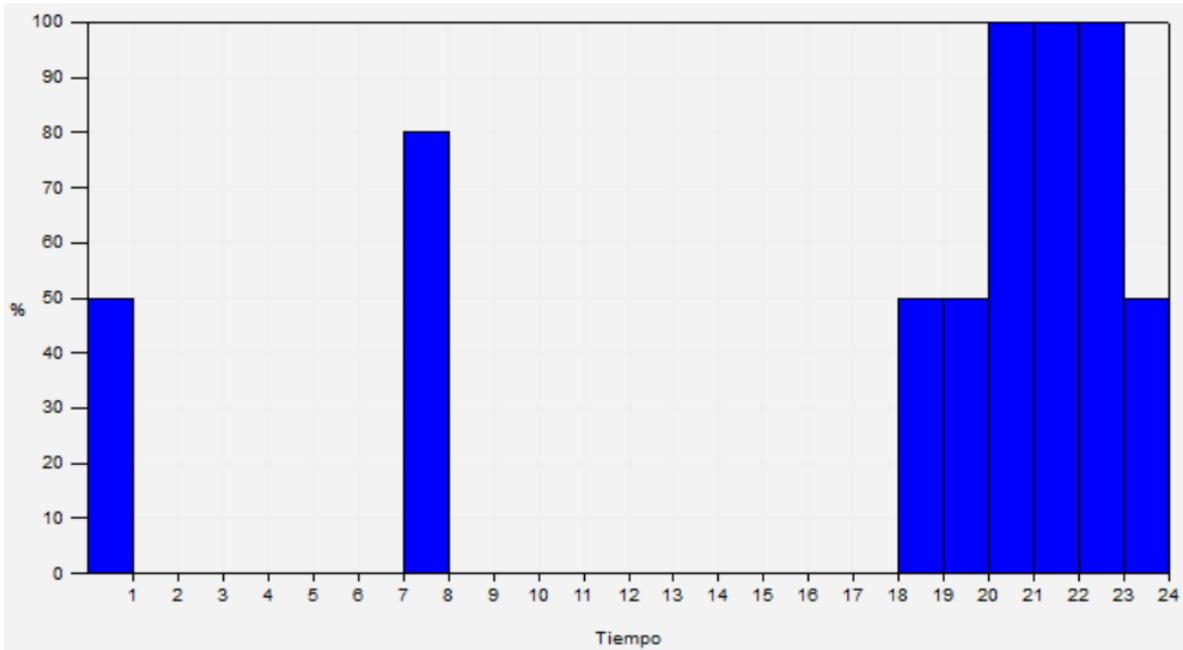


Anexo 3, Imagen 4: Perfil de Ocupación: Aseos

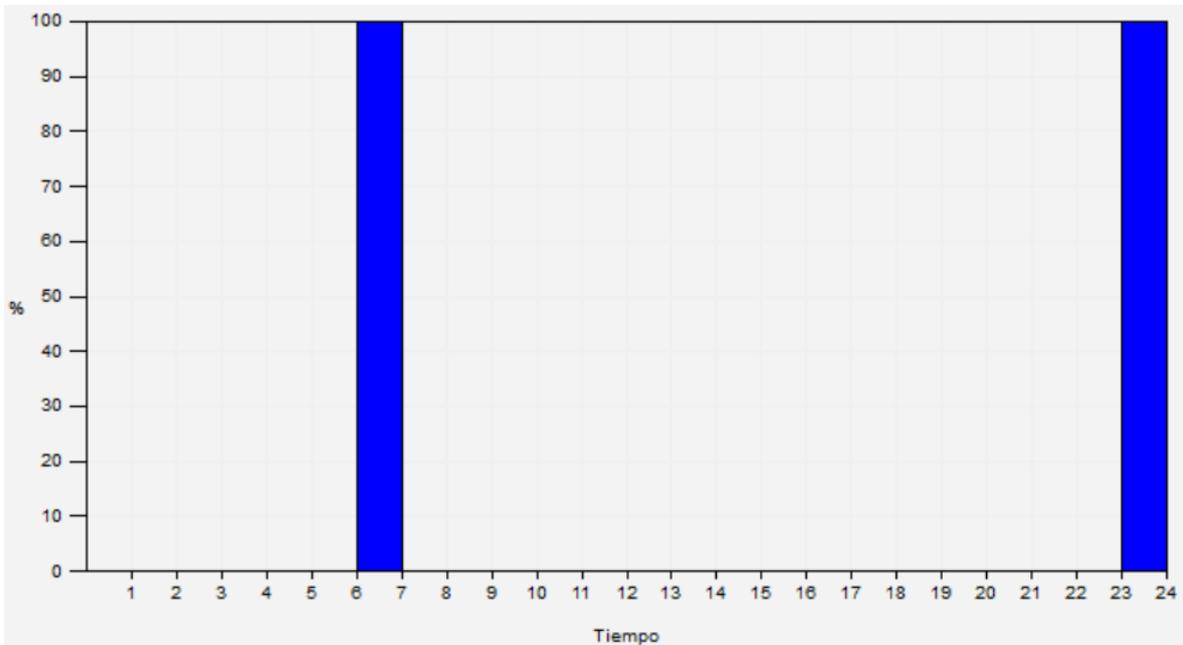


Anexo 3, Imagen 5: Perfil de Ocupación: Zona Común

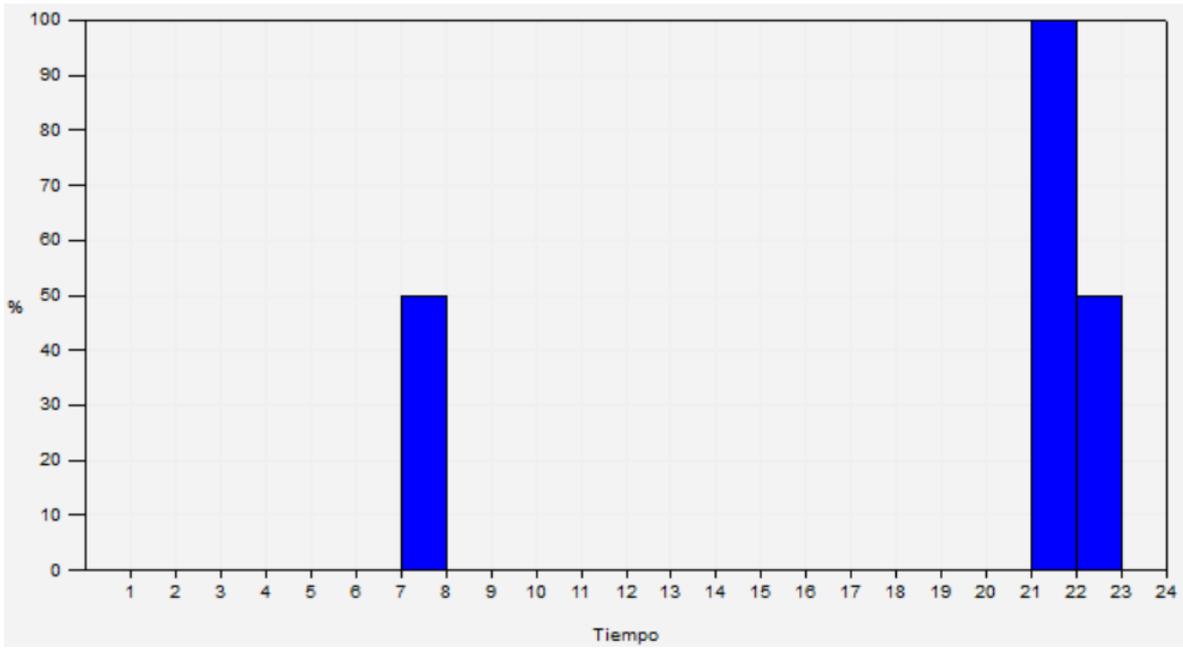
## ANEXO 6: Perfiles de Uso de la Iluminación



Anexo 4, Imagen 1: Perfil de Uso de la Iluminación: Zona de Estar y Cocina



Anexo 4, Imagen 2: Perfil de Uso de la Iluminación: Dormitorios



Anexo 4, Imagen 3: Perfil de Uso de la Iluminación: Iluminación de Tarea y Acento



# 06

## SCOP en Clima Medio y Cálido

		SPF = SCOPnet			
		CLIMA MEDIO		CLIMA CÁLIDO	
		35°	55°	35°	55°
Platinum BC iPlus	4,5 kW	4,83	3,44	6,87	4,68
	6 kW	4,53	3,52	6,06	4,61
Platinum iPlus V200 Integra	8 kW	4,55	3,49	5,87	4,47
Platinum BC iPlus V200 Smart	11 kW	4,56	3,40	5,85	4,37
	16 kW	4,47	3,33	5,71	4,17
Platinum BC iMax	22 kW	3,87	3,20	4,82	4,15
	27 kW	3,85	3,18	5,16	4,01
Platinum BC Plus Monobloc	5 kW	4,47	3,24	5,69	3,62
	6 kW	4,47	3,24	5,54	3,93
	7 kW	4,51	3,22	6,32	4,19
	12 kW	4,29	4,30	5,99	5,05
Platinum BC Monobloc Media Potencia Inverter	16 kW	3,23	3,27	4,45	3,93
	20 kW	3,61	3,28	4,74	3,66
	30 kW	4,00	3,26	5,25	4,13
	40 kW	4,36	3,47	5,78	4,41
Platinum BC Monobloc Media Potencia	25 kW	3,35	-	4,02	-
	38 kW	3,37	-	4,03	-

De acuerdo con la normativa UNE-EN 14825

## ANEXO 8: Código Python

### Sección 1: Módulo 'config.py'

```
import numpy as np
import pandas as pd
from modulos_tfg import coste_instalacion_anual

def selector_principal(df_consumo, ser_modulo, dataframes_list, rend, v1,v2,
                      precio_inv, periodo_explotacion, coste_montaje,
                      coste_estructura):

    # Inicialización de 3 variables con los datos del módulo elegido
    temp_coef = ser_modulo.loc['Coef. Pmax'] # Para Potencia (pu)
    pot_modulo_STC = ser_modulo.loc['Wp']    # Wp
    noct = ser_modulo.loc['NOTC (°C)']      # (°C) T.nominal

    # Inicialización de una variable con el precio del panel
    precio_panel = ser_modulo["Precio (€)"]

    # Inicialización de listas donde se almacenarán los resultados
    # de los cálculos para cada una de las orientaciones
    precio_ekWh_while = []
    n_placas_while = []
    temp_while_PV = []
    temp_while_E = []
    temp_while_coste = []

    # Inicialización de una variable contador
    k = 0

    # Recorrido de cada una de las orientaciones
    while k < len(dataframes_list):

        # Inicialización de listas donde se almacenarán los resultados de
        # los cálculos para cada N, siendo N el número de paneles.
        temp_for_precio = []
        temp_for_PV = []
        temp_for_E = []
        temp_for_coste = []

        print("Iteración número", k, "del bucle while")

        # Inicialización de un DF donde se almacenarán los resultados de la
        # potencia entregada por la instalación
        df_reslt_PV = pd.DataFrame()

        # Asignación del DF con la orientación a evaluar a df_meteo
        df_meteo = dataframes_list[k].copy()

        # Cálculo de la temperatura de las celdas en el momento t
        df_reslt_PV['T_cell'] = df_meteo['T2m'] + df_meteo['G(i)'] * (
            (noct - 20) / 800)
```

```

# Cálculo de la potencia entregada por cada placa en el momento t
df_reslt_PV['P_entregada'] = (df_meteo['G(i)'] * pot_modulo_STC *
                             np.exp((df_reslt_PV['T_cell'] - 25) *
temp_coef))

# Cambio de unidades
df_reslt_PV['P_entregada'] = df_reslt_PV['P_entregada'] * 1e-6

# Agregado de la columna de demanda eléctrica al DataFrame
df_reslt_PV['Demanda (kW)'] = df_consumo['Total (Electricidad)'].copy()

# Minoración de la potencia debido a las pérdidas en la instalación
df_reslt_PV['P_entregada'] = df_reslt_PV['P_entregada'] * rend

# Cálculos para N entre v1 y v2, siendo N el número de placas
for i in range(v1, v2):

    print("Iteracion numero", i, "del bucle while")

    # Copia del DF para evitar modificar el original y evitar fallos
    df_reslt_PV_copy = df_reslt_PV.copy()

    # Dada la P.entregada por placa en el momento t, cálculo de la
    # P.entregada por la instalación atendiendo a N y rendimiento
    df_reslt_PV_copy['P_entregada_inst'] =
df_reslt_PV_copy['P_entregada'] * i

    # Cálculo de la energía aprovechada
    E_aprovechada = (df_reslt_PV_copy[
        ["P_entregada_inst", "Demanda (kW)"]]).min(axis=1).sum()

    # Cálculo del coste anual asociado a equipos y montaje
    coste_inst = coste_instalacion_anual(
        i,
        precio_panel,
        coste_estructura,
        precio_inv,
        periodo_explotacion,
        coste_montaje
    )

    # Almacenamiento del DF con los cálculos para i placas
    temp_for_PV.append(df_reslt_PV_copy)
    # Almacenamiento de la energía aprovechada para i placas
    temp_for_E.append(E_aprovechada)
    # Almacenamiento del coste para i placas
    temp_for_coste.append(coste_inst)
    # Cálculo del precio del kWh para i placas
    precio_eWh = (coste_inst / E_aprovechada)
    # Adición del precio calculado a una lista
    temp_for_precio.append(precio_eWh)

```

```

# Obtención del precio mínimo
precio_ekWh_for = min(temp_for_precio)
# Obtención del índice del precio mínimo
idx_precio_ekWh_for = temp_for_precio.index(precio_ekWh_for)
# Obtención del número de placas que hace mínimo el precio
n_placas_for = idx_precio_ekWh_for + 1

""" En las líneas anteriores se han almacenado los datos del número de
placas que hace mínimo el precio. A continuación, se añaden estos
datos a otras listas, de forma que queden almacenados estos mismos
resultados, para cada una de las orientaciones """

# Adición a lista del precio del eWh elegido para la orientación k
precio_ekWh_while.append(precio_ekWh_for)
# Adición a lista del número de placas elegido para la orientación k
n_placas_while.append(n_placas_for)
# Adición a una lista del DF correspondiente
temp_while_PV.append(temp_for_PV[idx_precio_ekWh_for])
# Adición a lista del valor de la energía aprovechada correspondiente
temp_while_E.append(temp_for_E[idx_precio_ekWh_for])
# Adición a una lista del coste de la instalación PV correspondiente
temp_while_coste.append(temp_for_coste[idx_precio_ekWh_for])

# Actualización de la variable k (orientación)
k = k + 1

""" Obtención de los datos correspondientes al precio mínimo de las
4 orientaciones evaluadas """

# Obtención del precio (€/Wh) mínimo
precio_ekWh_final = min(precio_ekWh_while)
# Obtención del índice del precio mínimo
idx_orientacion = precio_ekWh_while.index(precio_ekWh_final)
# Obtención del número de placas
n_placas_final = n_placas_while[idx_orientacion]
# Obtención del DF con los cálculos
df_PV_final = temp_while_PV[idx_orientacion]
# Obtención de la energía aprovechada
E_aprov_final = temp_while_E[idx_orientacion]
# Obtención del coste
coste = round(temp_while_coste[idx_orientacion] * periodo_explotacion, 2)

# Retorno de DF con resultados
return precio_ekWh_final, n_placas_final, df_PV_final, idx_orientacion, \
    E_aprov_final, coste

```

## Sección 2. Módulo 'inversor.py'

```
import pandas as pd
import numpy as np

def selector_inversor(N, df_reslt_PV_sel, df_meteo, ser_modulo, df_inv, rend):

    # Temperatura mínima registrada
    Temp_min = df_meteo["T2m"].min()

    # Temperatura mínima minorada a efectos de seguridad
    Temp_min_seg = Temp_min - 5

    # Temperatura máxima registrada
    Temp_max = df_reslt_PV_sel["T_cell"].max()

    # Temperatura máxima mayorada a efectos de seguridad
    Temp_max_seg = Temp_max + 5

    # Potencia DC máxima esperada
    P_max_dc = (df_reslt_PV_sel["P_entregada_inst"] / rend).max()

    # Tensión mínima del módulo en circuito abierto en función de T
    V_oc_min_mod = ser_modulo["Voc (V)"] * (1 + ser_modulo["Coef. Voc"]
                                           * (Temp_max_seg - 25))

    # Tensión máxima del módulo en circuito abierto en función de T
    V_oc_max_mod = ser_modulo["Voc (V)"] * (1 + ser_modulo["Coef. Voc"]
                                           * (Temp_min_seg - 25))

    # Tensión mínima necesaria para la actuación del MPPT en función de T
    V_mpp_min_mod = ser_modulo["Vmp (V)"] * (1 + ser_modulo["Coef. Voc"]
                                              * (Temp_max_seg - 25))

    # Tensión máxima para la actuación del MPPT en función de T
    V_mpp_max_mod = ser_modulo["Vmp (V)"] * (1 + ser_modulo["Coef. Voc"]
                                              * (Temp_min_seg - 25))

    # Intensidad de cortocircuito del módulo en función de T
    I_sc_max_mod = ser_modulo["Isc (A)"] * (1 + ser_modulo["Coef. Isc"]
                                             * (Temp_max_seg - 25))

    # Intensidad de
    I_mpp_max_mod = ser_modulo["Imp (A)"] * (1 + ser_modulo["Coef. Isc"]
                                             * (Temp_max_seg - 25))

    # Tupla con las configuraciones / ramas en paralelo admitidas
    tupla_Np = tuple(range(1, N + 1))

    # Inicialización de un DF vacío en el cual se almacenarán los inversores
    # que cumplen los requisitos para cada configuración
    df_valid = pd.DataFrame()

    # Para cada una de las configuraciones de tupla_Np
    for i in tupla_Np:
        # Si no hay ramas en paralelo o las ramas están equilibradas
```

```

if i == 1 or N % 2 == 0:

    Np = i
    Ns = N / i

    # Valores de tensión y corriente en función de la configuración
    V_oc_max_inst = V_oc_max_mod * Ns
    V_oc_min_inst = V_oc_min_mod * Ns
    V_mpp_max_inst = V_mpp_max_mod * Ns
    V_mpp_min_inst = V_mpp_min_mod * Ns
    I_sc_max_inst = I_sc_max_mod * Np

    # Creación de una copia para evitar modificar el original
    df_inv_flt = df_inv.copy()

    # Se filtran los inversores que cumplen con los requisitos de
    # potencia, voltaje de circuito abierto, voltaje del punto de
    # máxima potencia y corriente de cortocircuito
    df_inv_flt = df_inv_flt[df_inv_flt["Pnom (kW)"] > P_max_dc]

    df_inv_flt = df_inv_flt[df_inv_flt["Udc_min (V)"] < V_oc_min_inst]

    df_inv_flt = df_inv_flt[df_inv_flt["Udc_max (V)"] > V_oc_max_inst]

    df_inv_flt = df_inv_flt[df_inv_flt["Vmpp_min (V)"] <
V_mpp_min_inst]

    df_inv_flt = df_inv_flt[df_inv_flt["Vmpp_max (V)"] >
V_mpp_max_inst]

    df_inv_flt = df_inv_flt[df_inv_flt["Idc_sc (A)"] > I_sc_max_inst]

    df_inv_flt.reset_index(inplace=True, drop=True)

    # Si hay inversores que cumplen todas las condiciones
if df_inv_flt.empty == False:
        # Adición de 2 columnas con información sobre la configuración
        df_inv_flt["Ns"] = Ns
        df_inv_flt["Np"] = Np
        # Tensión media
        df_inv_flt["Vmpp_med"] = (V_mpp_min_inst + V_mpp_max_inst) / 2

        df_inv_flt["prox_Vnom (V)"] = abs(df_inv_flt["Udc_nom(V)"] -
df_inv_flt["Vmpp_med"])

        df_inv_flt["sign"] = np.sign(df_inv_flt["Udc_nom(V)"] -
df_inv_flt["Vmpp_med"])

        # Adición de las filas correspondientes al DF
        df_valid = df_valid.append(df_inv_flt, ignore_index=True)

if not df_valid.empty:

    # Ordenación en función de la potencia nominal ,proximidad a V y precio
    df_valid = df_valid.sort_values(by=["Pnom (kW)", "prox_Vnom (V)",

```

```

                                "Precio (€)"])

    # Selección del inversor
    indice = df_valid["Pnom (kW)"].idxmin()
    Ns_sel = int(df_valid.loc[indice, "Ns"])
    Np_sel = int(df_valid.loc[indice, "Np"])

    return pd.Series(df_valid.loc[indice]), I_sc_max_mod * Np_sel, \
        I_mpp_max_mod * Np_sel, V_mpp_max_mod * Ns_sel, Ns_sel, \
        Np_sel - 1, Temp_max_seg

else:
    return pd.Series(dtype=int), 0, 0, 0, 0, 0, 0

```

### Sección 3. Módulo 'acs.py'

```

import pandas as pd
import numpy as np
from math import log, pi

def arreglo_acs(df_consumo):

    df_cons = df_consumo.copy()

    # Eliminación de la hora de la columna 'Fecha/Hora'
    df_cons['Fecha/Hora'] = df_cons['Fecha/Hora'].apply(lambda x: x.date())
    df_cons.rename(columns={'Fecha/Hora': 'Fecha'}, inplace=True)

    # La redistribución se hace dentro de un mismo día. Creación lista con
    # fechas
    lista_fechas = df_cons['Fecha'].unique().tolist()

    return df_cons, lista_fechas

def perdidas_deposito(r_int, espesor, k, em, L, T_agua, tramo, T_amb):

    # Inicialización de una variable T_sup para comenzar con proceso iterativo
    T_sup = T_amb + 5

    # Constante de Stefan Boltzmann
    cte_boltz = 5.67e-8

    # Radio exterior (Radio interior + espesor del aislante)
    r_ext = r_int + espesor

    # Resistencia térmica conductiva para una capa cilíndrica
    R_capa = log(r_ext / r_int) / (2 * pi * k)

    # Determinación de los parámetros para la iteración
    tolerancia = 1e-3

```

```

max_iteraciones = 100

for i in range(max_iteraciones):

    # Estimación inicial para comenzar el proceso iterativo
    T_sup_prev = T_sup

    # Cálculo del coeficiente de convección
    h_conv = 1.32 * ((T_sup - T_amb) / (r_ext * 2)) ** (1 / 4)

    # Cálculo del coeficiente de radiación
    h_rad = em * cte_boltz * (T_sup + T_amb) * (T_sup ** 2 + T_amb ** 2)

    # Coeficiente de convección / radiación
    h_conv_rad = h_conv + h_rad

    # Cálculo de la resistencia térmica por convección de capa cilíndrica
    R_conv_cil = 1 / (2 * pi * r_ext * h_conv_rad)

    # Cálculo de la resistencia térmica total
    R_total = R_conv_cil + R_capa

    # Cálculo de la pérdida de calor por metro
    qH_cil = (T_agua - T_amb) / R_total

    # Flujo de calor del depósito
    q = (qH_cil / (2 * pi * r_int)) * (pi * r_int ** 2 + 2 * pi * r_int *
L\
                                     + pi * r_int ** 2)

    # Actualización de la temperatura de la superficie
    T_sup = T_agua - qH_cil * R_total

    if abs(T_sup - T_sup_prev) < tolerancia:

        # Retorno de las pérdidas térmicas en kWh / tramo. Si el tramo
        # es de una hora, el retorno es en kWh.
        return (q * 1e-3 * tramo)

return 0

def strat(df_cons_p, lista_reha_completa):
    lista_reha_series = pd.Series(lista_reha_completa, index=df_cons_p.index)

    mask calefaccion = df_cons_p['Calefacción (Electricidad)'] > 0
    df_cons_p.loc[mask_calefaccion, 'Calefacción (Electricidad)'] -=
lista_reha_series[
    mask_calefaccion]

```

```

df_cons_p.loc[mask calefaccion, 'ACS (Electricidad)'] = (
    df_cons_p.loc[mask calefaccion, "ACS redistribuido (S.P)"]
    + lista_reha_series[mask calefaccion]
)

mask_refrigeracion = df_cons_p['Refrigeración (Electricidad)'] > 0
df_cons_p.loc[mask_refrigeracion, 'Refrigeración (Electricidad)'] +=
lista_reha_series[
    mask_refrigeracion]

df_cons_p.loc[mask_refrigeracion, 'ACS (Electricidad)'] = (
    df_cons_p.loc[mask_refrigeracion, "ACS redistribuido (S.P)"]
    + lista_reha_series[mask_refrigeracion]
)

mask_else = ~(mask calefaccion | mask_refrigeracion)
df_cons_p.loc[mask_else, 'ACS (Electricidad)'] = df_cons_p.loc[
    mask_else, "ACS redistribuido (S.P)"] + lista_reha_series[mask_else]

return df_cons_p

def redistribuidor_acs(df_cons, lista_fechas, df_rslt_PV_sel, args_fijos,
    df_estancia_acum):

    df_cons_p = df_cons.copy()

    # Cálculo del excedente para cada momento t (+ = excedente // - = deficit)
    df_cons["Dif."] = df_rslt_PV_sel["P_entregada_inst"] - df_cons[
        "Total (Electricidad)"]

    df_cons_p["Dif."] = df_rslt_PV_sel["P_entregada_inst"] - df_cons[
        "Total (Electricidad)"]

    # Inicialización de una matriz nula 48x48. Almacenará información sobre
    # de qué momento a qué momento se ha transferido la el consumo de energía.

    lista_ud_completa, lista_reha_completa = [], []

    # Para cada fecha contenida en lista_fechas
    for i in lista_fechas:

        matriz = np.zeros((48,48))

        # Inicialización de un DF reducido con los datos de consumo de la fecha
i
        datos_dia_i = df_cons_p[df_cons_p["Fecha"] == i]

        # Creación de un DF reducido con los datos de consumo de la fecha i

```

```

datos2_dia_i = df_estancia_acum[df_estancia_acum["Fecha"] == i]

# Reseteo de los índices dada la necesidad por utilización de la matriz
datos_dia_i.reset_index(inplace=True)
datos2_dia_i.reset_index(inplace=True)

# Inicialización de un vector con los excedentes de energía
ex = np.where(datos_dia_i["Dif."] > 0, datos_dia_i["Dif."],0)

# Inicialización de un vector con los déficits y cambio a signo (+)
defc = np.where(datos_dia_i["Dif."] < 0, datos_dia_i["Dif."], 0)
defc = - defc

# Inicialización de un vector con los datos de consumo de ACS
acs_in = np.array(datos_dia_i["ACS (Electricidad)"])
# Inicialización de un vector con el consumo de ACS no cubierto por PV
acs = np.minimum(defc, acs_in)
# Inicialización de un vector con la diferencia entre los 2 anteriores
# ya que de este consumo no se hará redistribución por estar cubierto
acs_dif = acs_in - acs

# Obtención del índice del primer momento de excedente
index_ex = np.argmax(ex!=0)

"""Variable auxiliar. Almacenará la producción desplazada y sobre
la que no se hace redistribución por t_ex > t_consumo"""
# Copia del vector acs
aux_acs = acs.copy()
# Truncado: establece a cero los elementos posteriores al 1º excedente
aux_acs[index_ex:] = 0

# Truncado: establece a cero los elementos anteriores al 1º excedente
acs[:index_ex] = 0

# Limitación del tiempo que el ACS puede permanecer almacenado
distancia_t_max = 10

# Inicialización de un bucle while
while np.sum(ex) > 0 and np.sum(acs) > 0:

    # Inicialización de los índices del par de elementos más cercano
    index_acs = np.argmax(acs!=0)
    index_ex = index_acs - np.argmax(ex[:index_acs + 1][::-1] != 0)

    # Interrupción del bucle si no queda ningún momento de excedente
    # antes de los momentos de producción de acs
    if index_acs == index_ex:
        break

    # Interrupción del bucle si se supera la "distancia" límite
    if abs(index_acs - index_ex) > distancia_t_max:
        break

    # Reasignación de los consumos asociados a la producción de ACS
    if ex[index_ex] > acs[index_acs]:

```

```

matriz[index_acs,index_ex] = acs[index_acs]
ex[index_ex] = ex[index_ex] - acs[index_acs]
aux_acs[index_ex] = aux_acs[index_ex] + acs[index_acs]
acs[index_acs] = 0

else:

matriz[index_acs,index_ex] = ex[index_ex]
acs[index_acs] = acs[index_acs] - ex[index_ex]
aux_acs[index_ex] = aux_acs[index_ex] + ex[index_ex]
ex[index_ex] = 0

# Suma de las listas para obtener
lista_acs_fin = acs + aux_acs + acs_dif
# Extensión a una lista con el periodo completo
lista_ud_completa.extend(lista_acs_fin)

""" Proceso de adición de pérdidas por desplazamiento """

# Inicialización de dos listas con 48 ceros
lista_reha = [0] * 48

# Obtención de una tupla con los índices de los elementos != 0
indexes = np.nonzero(matriz)

# Cálculo del volumen del depósito ( $V = \pi * r^2 * h$ )
volumen = pi * args_fijos[0]**2 * args_fijos[4] # m^3
# Cálculo de la masa de agua aprox. (con  $p=1000 \text{ kg/m}^3$ ) del depósito
masa = volumen * 1000 # kg
# Obtención del scop del depósito de los args. pasados por parámetro
Scop = args_fijos[8]
# Calor específico del agua
Cp = 4.186 # kJ/KgK

# Desempaquetado de la tupla
for m, n in zip(*indexes):
    print(matriz[m,n])
    k = m - 1

    while k >= n:

        # Cálculo de la energía térmica asociada a la producción de ACS
        # desplazada
        energia_kwh = matriz[m,n] * Scop
        # Cambio de unidades (kWh --> kJ)
        energia_kj = energia_kwh * 3600 # kJ

        # Obtención del incremento de temperatura logrado
        delta_T = energia_kj / (masa * Cp)
        # Obtención de la temperatura del depósito tras el aporte
        Tf = args_fijos[5] + delta_T

        # Cálculo de la temperatura promedio de la sala durante el
        # tiempo de almacenamiento

```

```

T_prom = datos2_dia_i.loc[n:m, "Temperatura del Aire"].mean()

# Cálculo de las pérdidas del depósito a la temperatura inicial
perdidas_Ti = perdidas_deposito(*(args_fijos + [T_prom]))

# Creación de una lista para la función, con la Temp. final
args_fijos_mod = args_fijos.copy()
args_fijos_mod[5] = Tf

# Cálculo de las pérdidas del depósito a la temperatura final
perdidas_Tf = perdidas_deposito(*(args_fijos_mod + [T_prom]))

# Cálculo de las pérdidas debido al incremento de temperatura
perdidas = perdidas_Tf - perdidas_Ti

# Obtención del consumo eléctrico adicional
consumo_e_adicional = perdidas / Scop

# Actualización de la lista
lista_reha[k] = lista_reha[k] + consumo_e_adicional

k = k - 1

# Extensión de la lista con el consumo adicional al periodo completo
lista_reha_completa.extend(lista_reha)

```

## Sección 4: Módulo 'electrico.py'

```

import pandas as pd
import numpy as np

def calcular_S(metodo, aislante, L, I_max, I_mpp_inst, V_mpp_inst, T_amb_max,
              f_corr=1):

    # Lectura y conversión a DF de las bases de datos necesarias
    i_adm1 = pd.read_excel("i_admisibles_cobre.xlsx", sheet_name="Hoja1")
    i_adm2 = pd.read_excel("i_admisibles_cobre.xlsx", sheet_name="Hoja2")

    # Obtención del índice columna correspondiente al método y aislante
    # indicados
    row_idx = i_adm1[i_adm1[1] == metodo].index
    col = (i_adm1.loc[row_idx, :] == aislante).T
    col_idx = col.idxmax()

    # Obtención del factor de corrección por temperatura haciendo uso de la
    # tabla e interpolando
    f_corr_T_DB = pd.read_excel("i_admisibles_cobre.xlsx", sheet_name="Hoja3")
    vap_aislante = ["XLPE" if "XLPE" in aislante else "PVC"]
    f_corr_T_DB = f_corr_T_DB[["T", vap_aislante[0]]]
    f_corr_T = np.interp(T_amb_max, f_corr_T_DB["T"],
                        f_corr_T_DB[vap_aislante[0]])

```

```

# Mayoración por seguridad de la corriente máxima en un 25%, y aplicación
# de los factores de corrección
I_max_seg = I_max * 1.25 * f_corr_T * f_corr

# Inicialización de las variables caída de tensión y una variable contadora
cdt = 100
cont = 0

# Mientras la caída de tensión porcentual sea mayor que 1.5
while cdt > 1.5:

    i = 0

    # Recorrido por las intensidades correspondientes a cada sección
    # hasta que se cumpla la condición especificada
    while i < len(i_adm2[col_idx]):
        if i_adm2.loc[i + cont, col_idx].values[0] > I_max_seg:
            S = i_adm2.loc[i + cont, 1]
            I_max_cable = i_adm2.loc[i + cont, col_idx].values[0]
            break

        i = i + 1

    # Inicialización de la variable correspondiente a la temperatura máxima
    # admisible en servicio permanente en función del tipo de aislante
    if "XLPE" in aislante:
        T_max_serv_perm = 90
    elif "PVC" in aislante:
        T_max_serv_perm = 70

    # Cálculo de la temperatura del cable en función de T y la corriente
    T_servicio = T_amb_max + (T_max_serv_perm - T_amb_max) * (
        I_mpp_inst / I_max_cable)**2

    # Coeficientes de resistividad térmica y eléctrica del cobre a T = 20
    °C
    alpha_cobre_20 = 0.00392
    p_cobre_20 = 0.018

    # Corrección por temperatura
    p_cobre_T = p_cobre_20 * (1 + alpha_cobre_20 * (T_servicio - 20))

    # Cálculo de la caída de tensión
    cdt = (p_cobre_T * L * I_mpp_inst * 100) / (S * V_mpp_inst)
    R = p_cobre_T * L / S

    cont = cont + 1

# Retorno de la sección, la intensidad máxima admitida por el cable y de
cdt
return S, I_max_cable, cdt, R

```

## Sección 5: Módulo 'modulos\_tfg.py'

```
import os
import glob
import pandas as pd

def procesar_archivos(consumo_compartido, ruta):

    # Comprobación de la variable consumo_compartido
    if consumo_compartido == "N":
        # Lista todos los archivos csv en la carpeta
        archivos = glob.glob(os.path.join(ruta, "*.csv"))

        # Si hay más de un archivo CSV en la carpeta, arroja un error
        if len(archivos) != 1:
            raise ValueError("Hay más de un archivo CSV en la carpeta.")

        else:
            # Lee el archivo CSV y lo convierte en un DataFrame
            df_consumo_fix = pd.read_csv(archivos[0],
                                        sep=";",
                                        encoding="ISO-8859-1",
                                        decimal=",")

            return df_consumo_fix

    elif consumo_compartido == "S":
        # Lista todos los archivos csv en la carpeta
        archivos = glob.glob(os.path.join(ruta, "*.csv"))

        # Inicializa el DataFrame
        df_consumo_fix = pd.DataFrame()

        # Para cada archivo en la carpeta
        for archivo in archivos:
            # Lee el archivo CSV
            df_temp = pd.read_csv(archivo,
                                  sep=";",
                                  encoding="ISO-8859-1",
                                  decimal=",")

            # Si df_consumo_fix está vacío, copia el primer DataFrame
            if df_consumo_fix.empty:
                df_consumo_fix = df_temp

            else:
                # Comprueba si la primera columna coincide
                if not df_consumo_fix.iloc[:,0].equals(df_temp.iloc[:,0]):
                    raise ValueError("Las fechas no coinciden en los
archivos.")

                # Suma las columnas (excepto la primera)
                df_consumo_fix.iloc[:,1:] += df_temp.iloc[:,1:]

        return df_consumo_fix
```

```

    else:
        raise ValueError("El valor de 'consumo_compartido' debe ser 'N' o
'S'.")

def coste_instalacion_anual(n_paneles, precio_panel, coste_estructura,
                            precio_inversor, periodo_explotacion,
                            coste_montaje):

    coste_equip = n_paneles * (precio_panel + coste_estructura) +
precio_inversor

    coste_total = coste_equip * (1 + coste_montaje)

    coste_total_anual = coste_total / periodo_explotacion

    return coste_total_anual

def perdidas_modulo(ser_modulo, periodo_explotacion):

    # Cálculo de las pérdidas del módulo asociadas al envejecimiento

    from statistics import mean

    year1_degradation = 1 - ser_modulo["Degradación_1"]
    rest_degradation = ser_modulo["Degradación_resto"]

    eta_anual = list(range(periodo_explotacion))
    eta_anual[0] = year1_degradation

    i = 1
    while i < len(eta_anual):
        eta_anual[i] = eta_anual[i - 1] - rest_degradation

        i = i + 1

    return mean(eta_anual)

def rend_inst(ser_modulo, ser_inv, R_DC, R_AC, df_meteo_sel, Vmpp_med,
              periodo_explotacion):

    # Tolerancia y envejecimiento
    Pper_1 = 1 - perdidas_modulo(ser_modulo, periodo_explotacion)
    # Mismatch
    Pper_2 = 0.03
    # Polvo y suciedad
    Pper_3 = 0.03

```

```

# Sombras
Pper_4 = 0.01
# Perdidas panel
Pper_panel = Pper_1 + Pper_2 + Pper_3 + Pper_4
# Inversor
Pper_inv_1 = 1 - ser_inv["eta_EU"]

#=====

# Pérdidas en el inversor
## Pendientes de las curvas de rendimiento EU en función del voltaje DC
m1 = 0.01111
m2 = 0.00357

if ser_inv["sign"] == 1:
    Pper_inv_2 = ser_inv["prox_Vnom (V)"] * m1 / 100
else:
    Pper_inv_2 = ser_inv["prox_Vnom (V)"] * m2 / 100

Pper_inv = Pper_inv_1 + Pper_inv_2

#=====

# Conductores DC
df_meteo_sel["Impp_med_DC"] = df_meteo_sel["P_entregada_inst"] * (
    1 - Pper_panel) / (Vmpp_med * 1e-3)

df_meteo_sel["P_perd_DC"] = df_meteo_sel["Impp_med_DC"]**2 * R_DC * 1e-3

df_meteo_sel["P_perd_DC_porcent"] = (
    df_meteo_sel.loc[:, "P_perd_DC"] * 100
) / df_meteo_sel.loc[:, "P_entregada_inst"]

# Potencia de pérdidas en el cableado DC
Pper_cond_DC = (df_meteo_sel["P_perd_DC_porcent"] / 100).mean()

#=====

# Conductores AC
df_meteo_sel["P_sal_inv"] = df_meteo_sel["P_entregada_inst"] * (
    1 - Pper_inv - Pper_panel)

df_meteo_sel["Impp_med_AC"] = df_meteo_sel["P_sal_inv"] / 0.230 # kV
df_meteo_sel["P_perd_AC"] = df_meteo_sel["Impp_med_AC"]**2 * R_AC * 1e-3

df_meteo_sel["P_perd_AC_porcent"] = (
    df_meteo_sel["P_perd_AC"] * 100) / df_meteo_sel["P_sal_inv"]

# Potencia de pérdidas en el cableado AC
Pper_cond_AC = (df_meteo_sel["P_perd_DC_porcent"] / 100).mean()

#=====

# Otros
Pper_rest = 0.02

# Cálculo del rendimiento de la instalación

```

```

rend = 1 - (Pper_panel + Pper_inv + Pper_cond_DC + Pper_cond_AC +
Pper_rest)

return rend

def calcular_importe(fila, precio_ekWh_inst, precio_ekWh_red):

    # Si no hay excedente
    if fila['Dif.'] > 0:
        return fila['Total (Electricidad)'] * precio_ekWh_inst

    # Si hay excedente
    elif fila['Dif.'] <= 0:
        return - fila['Dif.'] * precio_ekWh_red + (
            fila['Total (Electricidad)'] + fila['Dif.']) * precio_ekWh_inst

def distancia(ser_modulo, inclinacion, latitud):

    from math import cos, radians, tan

    altura_modulo_inclinado = ser_modulo["L (mm)"] * cos(radians(inclinacion))
    k = 1 / tan(radians(latitud))

    return altura_modulo_inclinado * k * 1e-3

# Pasamos los valores de temperatura de °C a K
def pasar_a_K(columna):
    if columna.name != 'Fecha/Hora':
        return columna + 273
    return columna

```

## Sección 6: Módulo 'simulador.py'

```

import os
import re
import pandas as pd
import numpy as np

def calcular_reparto(ruta_carpeta, p_entregada):

    # Inicialización de una lista con los nombres contenidos en la carpeta
    archivos = os.listdir(ruta_carpeta)

    # Inicialización de un DataFrame vacío para almacenar los datos
    df_total = pd.DataFrame()

    # Adición de una columna con la potencia entregada por la instalación
    df_total['P_entregada_inst'] = p_entregada

```

```

# Inicialización de una lista donde se almacenarán los números de las
# viviendas
lista_num = []

# Iteración sobre cada archivo en la carpeta
for archivo in archivos:

    # Extracción del número correspondiente a la casa
    num = re.findall(r'\d', archivo)
    # Adición del anterior a lista_num
    lista_num.append(''.join(num))

#
if archivo.endswith('.csv'):
    # Creación de un DataFrame a partir del archivo CSV
    df = pd.read_csv(os.path.join(ruta_carpeta, archivo),
                      sep=";",
                      encoding="ISO-8859-1",
                      decimal=",")

    # Construcción de una cadena con el número de la vivienda a
    # continuación del consumo eléctrico, de forma que se los
identifique
    cadena = 'Total (Electricidad)' + ' ' + ''.join(num)

    # Agregación de los datos del archivo al DataFrame total
    df_total[cadena] = df['Total (Electricidad)']

# Creación de una nueva columna con el consumo sumado de todas las
viviendas
df_total['Consumo conjunto'] = df_total.iloc[:, 1:].sum(axis=1)

""" Creación de nuevas columnas en las cuales se almacenará el porcentaje
de demanda respecto del total para cada casa e instante"""

# Solo se consideran los instantes en los que hay producción PV
mask = df_total['P_entregada_inst'] > 0

for numero in lista_num:
    # Creación de la cadena con el nombre de la columna correspondiente a
la
    # iteración en curso
    cadena = '% Casa' + ' ' + numero

    # Creación de una nueva columna y relleno de las celdas con
    # el porcentaje de demanda sobre el total
    df_total.loc[mask, cadena] = df_total.loc[mask, 'Total (Electricidad)'
+ ' ' + numero] \
    / df_total.loc[mask, 'Consumo conjunto']

    # En caso de que no hubiera producción PV el porcentaje se fija en cero
    df_total.loc[~mask, cadena] = 0

```

```

# Obtención del mínimo entre producción PV y demanda, es decir, la
# cantidad que se va a aprovechar
df_total["min"] = df_total.apply(
    lambda x: min(x["P_entregada_inst"], x["Consumo conjunto"]), axis=1)

# Creación de una lista donde se almacenará el % de energía que
# corresponderá a cada casa y por tanto, el porcentaje que le corresponderá
# pagar del coste total de la instalación

asignacion = []

for numero in lista_num:
    cadena = 'Reparto' + ' ' + numero
    df_total[cadena] = df_total["min"] * df_total['% Casa' + ' ' + numero]

for numero in lista_num:
    cadena = 'Reparto' + ' ' + numero
    fraccion = df_total[cadena].sum() / df_total['min'].sum()
    asignacion.append(round(fraccion, 2))

l_apoyo = [x for x in df_total if 'Total (Electricidad)' in x]
df_final = pd.DataFrame(p_entregada)
df_final = pd.concat([df_final["P_entregada_inst"], df_total.loc[:,
l_apoyo]]
                    , axis=1)

for numero in lista_num:
    cadena = 'P_asignada' + ' ' + numero
    df_final[cadena] = df_final['P_entregada_inst'] *
asignacion[int(numero) - 1]

return df_final, asignacion, lista_num

def repartir_excedentes(row, asignacion, lista_num):

    asignacion = np.array(asignacion)
    consumo = [row['Total (Electricidad)' + ' ' + n] for n in lista_num]
    reparto_inicial = row['P_entregada_inst'] * asignacion
    consumo_por_cubrir = np.maximum(consumo - reparto_inicial, 0)
    excedente_inicial = np.maximum(reparto_inicial - consumo, 0)

    suma_excedente = np.sum(excedente_inicial)
    suma_sincubrir = np.sum(consumo_por_cubrir)
    a_reasignar = np.minimum(suma_sincubrir, suma_excedente)

    coef_reasignacion = np.zeros(len(lista_num))
    coef_reasignacion_ex = np.zeros(len(lista_num))

    if a_reasignar != 0:

```

```

coef_reasignacion = consumo_por_cubrir / suma_sincubrir
coef_reasignacion_ex = excedente_inicial / suma_excedente

# Lista con lo que cede cada casa
ex_fin = a_reasignar * coef_reasignacion_ex
# Lista con el excedente final. El inicial menos el que ha cedido cada casa
excedente_final = excedente_inicial - ex_fin
# Lista con lo que recibe cada casa
ex_reasignado = a_reasignar * coef_reasignacion
# Lista con el nuevo reparto de potencia
reparto_con_ex = reparto_inicial + ex_reasignado - ex_fin

return reparto_con_ex, excedente_final

```

## Sección 7: Módulo 'principal.py'

```

import pandas as pd
import numpy as np

from config import selector_principal
from inversor import selector_inversor
from acs import redistribuidor_acs, arreglo_acs
from modulos_tfg import rend_inst, distancia, procesar_archivos
from modulos_tfg import calcular_importe, pasar_a_K
from electrico import calcular_S
from simulador import calcular_reparto, repartir_excedentes
from impresion import imprimir, imprimir_reslt_energia, imprimir_reslt_acs

#=====

# Seleccionar si se va a dimensionar la instalación de forma conjunta (S / N)
dimensionado_conjunto = "S"
# Seleccionar si se va a simular la redistribución de acs (S / N)
modo_acs = "N"
# Seleccionar si se va a simular la compartición de excedentes (S / N)
modo_simulacion = "S"
# Seleccionar el número de casa sobre el cual se desea analizar el impacto
# de la compartición de energía
n_casa = 1
#=====

# Latitud (Grados)
latitud = 41.6
# Inclinación del panel (Grados)

```

```

inclinacion = 41.6

#=====

# Periodo de explotación o vida útil (Años)
periodo_explotacion = 25
# Precio promedio estimado de la energía de la red (€/kWh)
precio_ekWh_red = 0.27
# Incremento estimado del coste de la instalación debido al montaje
coste_montaje = 0.2
# Coste de la estructura de cada panel (€)
coste_estructura = 50

#=====

# Longitud del cable del tramo de continua [Paneles --> Inversor] (m)
longitud_cable_dc = 20
# Longitud del cable del tramo de alterna [Inversor --> Cuadro] (m)
longitud_cable_ac = 5
# Método de instalación del tramo de continua ( A1/ A2 / B1 / B2 / C / E / F )
metodo_cable_dc = "C"
# Método de instalación del tramo de alterna ( A1/ A2 / B1 / B2 / C / E / F )
metodo_cable_ac = "C"
# Aislante empleado para el tramo de continua ( PVC2 / PVC3 / XLPE2 / XLPE3 )
aislante_dc = "XLPE2"
# Aislante empleado para el tramo de alterna ( PVC2 / PVC3 / XLPE2 / XLPE3 )
aislante_ac = "XLPE3"
# Factor de corrección total, aparte del de temperatura. 1 por defecto.
f_corr_DC = 1
f_corr_AC = 1

#=====

# Radio interior del depósito
r_int = 0.19
# Espesor del aislante
espesor = 0.04 #m
# Conductividad del aislante
k = 0.022 # W/mK
# Emisividad de la superficie del depósito
em = 0.9
# Longitud del depósito (m)
L = 1
# Temperatura del agua del depósito (K)
T_agua = 60 + 273
# Intervalos de tiempo (En este caso particular (TFG) de media hora)
tramo = 0.5
# SCOP
SCOP = 4.5

#=====

# Ruta de la carpeta con los consumos
ruta = r"C:\Users\andre\OneDrive\Escritorio\Fotovoltaica\Consumo"

df_consumo = procesar_archivos(dimensionado_conjunto, ruta)

```

```

# Conversión de la columna 'Fecha/Hora' a Datetime
df_consumo['Fecha/Hora'] = pd.to_datetime(df_consumo['Fecha/Hora'],
                                         infer_datetime_format=True)

# Redondeo a fin de evitar demasiados decimales
df_consumo = df_consumo.apply(
    lambda x: x.round(4) if x.dtype == 'float64' else x)

#=====

# Creación de una lista de nombres de las bases de datos meteorológicas
file_list = ['database_deg_min_30', 'database_deg_min_15',
             'database_deg_0', 'database_deg15', 'database_deg30']

# Creación de una lista vacía donde se almacenarán los dataframe leídos
dataframes_list = []

# Recorrido de la lista de archivos y creación de un DF por cada archivo
for i in range(len(file_list)):
    # Leemos cada archivo y lo convertimos a dataframe
    temp_df = pd.read_csv(file_list[i]+'.csv')
    # Añadimos el dataframe a la lista
    dataframes_list.append(temp_df)

#=====

# Importación y conversión a DF de la base de datos de paneles fotovoltaicos
df_modulos = pd.read_excel('database_modulos.xlsx')

# Agrupación por características similares y extracción de un modulo por grupo
df_modulos_group = df_modulos.groupby("Ficha", as_index=False)
df_modulos_simp = df_modulos_group.first()

#=====

# Lectura y conversión a DF de la base de datos de los inversores
df_inv = pd.read_excel('database_inversores.xlsx')

#=====

# Inicialización de varias listas en las cuales se almacenarán los resultados
rslt_sel_principal, rslt_elec_DC, rslt_elec_AC = [], [], []
rslt_rend, rslt_inverter = [], []

#=====

# Recorrido de df_modulos_simp
for i in range(len(df_modulos_simp)):

    # Imprime por pantalla el módulo para el cual se está diseñando la
    # instalación
    print("Evaluando el módulo número", i)

```

```

# Asignación del módulo correspondiente a la variable modulo
modulo = df_modulos_simp.iloc[i, :]

# Asignación de un precio medio para el inversor para la primera iteración
precio_inv = df_inv['Precio (€)'].mean()

""" CREO QUE NO ES NECESARIO REALIZAR ESTA COPIA """
# Para cada módulo se parte de la situación : ACS sin distribuir
#df_consumo = df_consumo_fix.copy()

# Valor de partida del rendimiento para el proceso iterativo
rendimiento = 0.86
# 1ª variable de control: Tolerancia permitida entre el precio (€/kWh)
# obtenido en la iteración i y el obtenido en la i-1
tolerancia = 0.001
# 2ª variable de control del proceso iterativo
contador = 0
# Flag de salida para el proceso iterativo
condicion = True

# Establecimiento de variables de comparación
prev_precio = 0
prev_num_modulos = 0

# Limitación del rango para la función selector_principal
v1 = 1 # No cambia
v2 = 20 # Puede hacersele variar para agilizar la ejecución

""" ESTA EXPLICACION NO ES CLARA """
# Simplemente las inicializamos para que el "visualizador no de error"
lista_f_prep_acs, df_meteo_sel = [], []

while condicion:

    print("Iteración número", contador)

    # Llamado a la función selector_principal del módulo config
    precio, num_modulos, df_reslt_PV_sel, idx, \
        E_aprovechada, coste_inst = selector_principal(
        df_consumo,
        modulo,
        dataframes_list,
        rendimiento,
        v1,
        v2,
        precio_inv,
        periodo_explotacion,
        coste_montaje,
        coste_estructura
    )

    # Llamado a la función selector_inversor del módulo inverter

```

```

        ser_inv, I_max_DC, I_mpp_max_DC, V_mpp_max_DC, Ns, Np, T =
selector_inversor(
    num_modulos,
    df_reslt_PV_sel,
    dataframes_list[idx],
    modulo,
    df_inv,
    rendimiento
)

# Si no se ha encontrado un inversor que cumpla con los requisitos
if ser_inv.empty:

    print("No se ha encontrado inversor para esta configuración")

    rslt_sel_principal.append(None)
    rslt_inverter.append(None)
    rslt_elec_DC.append(None)
    rslt_elec_AC.append(None)
    rslt_rend.append(None)

    break

else:

    print("Se ha encontrado inversor para esta configuración")

    precio_inv = ser_inv['Precio (€)']
    I_max_AC = ser_inv['I_ac_max (A)']

    # Llamado a la función calcular_S con los datos correspondientes a
DC
    S_DC, I_max_cond_DC, cdt_DC, R_DC = calcular_S(
        metodo_cable_dc,
        aislante_dc,
        longitud_cable_dc,
        I_max_DC,
        I_mpp_max_DC,
        V_mpp_max_DC,
        T,
        f_corr_DC
    )

    # Llamado a la función calcular_S con los datos correspondientes a
AC
    S_AC, I_max_cond_AC, cdt_AC, R_AC = calcular_S(
        metodo_cable_ac,
        aislante_ac,
        longitud_cable_ac,
        I_max_AC,
        I_mpp_max_DC,
        230,
        T,
        f_corr_AC
    )

```

```

    )

    # Llamado a la función rend_inst del módulo modulos_tfg
    rendimiento = rend_inst(
        modulo,
        ser_inv,
        R_DC,
        R_AC,
        df_reslt_PV_sel,
        ser_inv['Vmpp_med'],
        periodo_explotacion
    )

    rendimiento = rendimiento.round(3)

    if condicion == False:
        break

    elif (abs(precio - prev_precio) < tolerancia) and (
        num_modulos == prev_num_modulos):
        break

    if contador > 9:

        minimo = min(precio, prev_precio)

        if minimo == precio:
            break

        else:
            condicion = False

        #v2 = num_modulos + 2
        prev_precio = precio
        prev_num_modulos = num_modulos
        contador = contador + 1

    if ser_inv.empty:
        continue

    else:

        # Se crean nuevas listas a las cuales se les van añadiendo los
        resultados
        # de los diferentes modulos
        rslt_sel_principal.append([precio, num_modulos, df_reslt_PV_sel, idx,
            E_aprovechada, coste_inst])

        rslt_inverter.append([ser_inv, I_max_DC, I_mpp_max_DC, V_mpp_max_DC,
            Ns, Np, T])

        rslt_elec_DC.append([S_DC, I_max_cond_DC, cdt_DC, R_DC])
        rslt_elec_AC.append([S_AC, I_max_cond_AC, cdt_AC, R_AC])
        rslt_rend.append(rendimiento)

```

```

# Almacenamiento en una lista del importe resultante de cada configuración
control_inversor = [sublista[0] if sublista is not None else None for sublista
in rslt_sel_principal]

# Obtención del importe mínimo exceptuando aquellos para los cuales no se
# ha encontrado inversor
result = min(l for l, f in zip(control_inversor, rslt_inverter) if f is not
None)

# Obtención del índice de result
indice = control_inversor.index(result)

rslt_sel_principal_sel = rslt_sel_principal[indice]
rslt_inverter_sel = rslt_inverter[indice]
rslt_elec_DC_sel = rslt_elec_DC[indice]
rslt_elec_AC_sel = rslt_elec_AC[indice]
rslt_rend_sel = rslt_rend[indice]
resultados = [rslt_sel_principal_sel, rslt_inverter_sel, rslt_elec_DC_sel,
rslt_elec_AC_sel, rslt_rend_sel]

""" Ojo este de bajo tambien esta en acs """
df_consumo["Dif."] = resultados[0][2]["P_entregada_inst"] - df_consumo[
"Total (Electricidad)"]

importe = df_consumo.apply(lambda fila: calcular_importe(
fila, resultados[0][0], precio_ekWh_red), axis=1).sum()

resultados.append(importe)

distancia = distancia(df_modulos_simp.loc[indice, :], latitud, latitud)
distancia = round(distancia, 2)
resultados.append(distancia)

#===== ACS =====

# Determinar si se va a ejecutar el módulo de acs

if modo_acs == "S" and dimensionado_conjunto == "S":

    raise Exception("El modo acs no está pensado para un dimensionado conjunto")

elif modo_acs == "S" and dimensionado_conjunto == "N":

    # Lectura y creación de un DF con los datos de temperatura de la
    # estancia en la que se encuentra el acumulador
    df_estancia_acum = pd.read_csv('Condiciones_estancia_acumulador.csv',
sep=";",
encoding="ISO-8859-1", decimal=",")

```

```

# Conversión de la columna 'Fecha/Hora' a Datetime
df_estancia_acum['Fecha/Hora'] =
pd.to_datetime(df_estancia_acum['Fecha/Hora'],
               infer_datetime_format=True)

# Llamado a la función 'pasar_a_K'
df_estancia_acum = df_estancia_acum.apply(pasar_a_K)

args_fijos = [r_int, espesor, k, em, L, T_agua, T_sup_inicial, tramo, SCOP]

# Preprocesado del DF de consumo para ser pasado al modulo principal de ACS
df_consumo_prep_acs, lista_f_prep_acs = arreglo_acs(df_consumo)

# Preprocesado del DF 'estancia_acum' para ser pasado al modulo principal
de ACS
df_acum_proc, l = arreglo_acs(df_estancia_acum)

# Llamado a la función redistribuidor_acs
df_cons, df_cons_acs, lista_q = redistribuidor_acs(
    df_consumo_prep_acs,
    lista_f_prep_acs,
    resultados[0][2],
    args_fijos,
    df_acum_proc,
    SCOP)

q1 = resultados[0][2]["P_entregada_inst"]
q2 = df_cons_acs["Total (Electricidad)"]

e_aprovechada_acs = np.minimum.reduce([q1, q2]).sum()

# Cálculo del precio del kWh
precio_ekWh_inst_acs = (resultados[0][5] / periodo_explotacion) /
e_aprovechada_acs

# Incremento del gasto energético debido a la redistribución en tanto por
uno
d_energia = (df_cons_acs['Total (Electricidad)'].sum() / \
             df_consumo['Total (Electricidad)'].sum()) - 1

importe_acs = df_cons_acs.apply(lambda fila: calcular_importe(
    fila, precio_ekWh_inst_acs, precio_ekWh_red), axis=1).sum()

else:
    pass

#=====

```

```

# Impresión de resultados
imprimir(resultados, df_modulos_simp, indice)
imprimir_reslt_energia(resultados, dimensionado_conjunto)

if modo_acs == "S":
    imprimir_reslt_acs(
        resultados,
        modo_acs,
        e_aprovechada_acs,
        d_energia,
        precio_ekWh_inst_acs,
        importe_acs
    )

if dimensionado_conjunto == 'S':

    print("----- RESULTADOS CASA DIMENSIONADO CONJUNTO -----")

    # Llamado a la función calcular_reparto
    df_cc, asignacion, lista_num = calcular_reparto(ruta,
resultados[0][2]['P_entregada_inst'])
    print("Coeficientes de reparto:", asignacion)

    # Inicialización de un nuevo DF para almacenar estos resultados
    df_casa = pd.DataFrame()
    # Adición de la potencia correspondiente a la casa analizada al DF
    df_casa["P_asignada"] = df_cc["P_asignada"] + " " + str(n_casa)]

    # Cálculo del importe correspondiente a la casa analizada
    coste_casa_cc = resultados[0][5] * asignacion[n_casa - 1]

    # Adición de su consumo a la casa analizada para poder utilizar
'calcular_importe'
    df_casa["Total (Electricidad)"] = df_cc["Total (Electricidad)"] + " " +
str(n_casa)]
    # Cálculo de la energía aprovechada por la casa analizada
    e_aprovechada_casa_cc = np.minimum.reduce([df_casa["P_asignada"],
df_casa["Total
(Electricidad)"]]).sum()

    # Cálculo del precio del precio del kWh para la casa analizada
    precio_ekWh_casa_cc = (coste_casa_cc / periodo_explotacion) /
e_aprovechada_casa_cc
    # Creación de una nueva columna para la función 'calcular_importe'
    df_casa["Dif."] = df_casa["P_asignada"] - df_casa["Total (Electricidad)"]
    # Llamado a 'calcular_importe'
    importe_casa_cc = df_casa.apply(lambda fila: calcular_importe(
        fila, precio_ekWh_casa_cc, precio_ekWh_red), axis=1).sum()

    # Impresión de resultados
    coste_casa_cc = round(coste_casa_cc, 2)

```

```

print("Coste de la instalación para la casa" + " " + str(n_casa),
      coste_casa_cc, "€")

precio_ekWh_casa_cc = round(precio_ekWh_casa_cc, 6)
print("Precio de la energía para la casa" + " " + str(n_casa),
      precio_ekWh_casa_cc, "€/kWh")

importe_casa_cc = round(importe_casa_cc, 2)
print("Importe de la factura eléctrica para la casa" + " " + str(n_casa) +
":",
      importe_casa_cc, "€")

if modo_simulacion == "S":

    print("----- RESULTADOS CASA SIMULACIÓN -----")

    nombres_p = ["P_asignada_sim" + " " + str(i) for i in lista_num]
    nombres_e = ["Excedente" + " " + str(i) for i in lista_num]
    nombres_t = ["Total (Electricidad)" + " " + str(i) for i in lista_num]

    df_cc[nombres_p] = df_cc.apply(
        lambda row: pd.Series(
            repartir_excedentes(row, asignacion, lista_num)[0]), axis=1)

    E_aprovechada_por_casa = []

    for p, t in zip(nombres_p, nombres_t):
        E_aprovechada = df_cc[[p, t]].min(axis=1).sum()
        E_aprovechada_por_casa.append(E_aprovechada)

    e_entregada = resultados[0][2]["P_entregada_inst"].sum()

    porc_e_aprovechada = (sum(E_aprovechada_por_casa) * 1e2) / e_entregada
    porc_e_aprovechada = round(porc_e_aprovechada, 2)
    print("% Energía aprovechada total:", porc_e_aprovechada, "%")

    # Cálculo del precio del kWh para la casa analizada
    precio_ekWh_casa_sim = (coste_casa_cc / periodo_explotacion) / (
        E_aprovechada_por_casa[n_casa - 1])
    precio_ekWh_casa_sim = round(precio_ekWh_casa_sim, 6)
    print("Precio de la energía para la casa" + " " + str(n_casa),
        precio_ekWh_casa_sim, "€/kWh")

    # Inicialización de un nuevo DF para almacenar estos resultados
    df_sim_casa = pd.DataFrame()
    # Adición de su consumo a la casa analizada para poder utilizar
    'calcular_importe'
    df_sim_casa["Total (Electricidad)"] = df_cc["Total (Electricidad)" +
        " " + str(n_casa)]

    # Creación de una nueva columna para la función 'calcular_importe'
    df_sim_casa["Dif."] = df_cc["P_asignada_sim" +
        " " + str(n_casa)] - df_sim_casa["Total
(Electricidad)"]

```

```
# Llamado a 'calcular_importe'
importe_sim_casa = df_sim_casa.apply(lambda fila: calcular_importe(
    fila, precio_ekWh_casa_sim, precio_ekWh_red), axis=1).sum()
importe_sim_casa = round(importe_sim_casa, 2)
print("Importe de la factura eléctrica para la casa" + " " + str(n_casa) +
":",
      importe_sim_casa, "€")
```

## ANEXO 9: Presupuesto

Cursos y Licencias:

- Curso “Aprende DesignBuilder” + Licencia: 359,37 €

Salario personal: 20 € / hora

Trabajo de oficina:

- Curso “Aprende DesignBuilder”: 10 h.
- Modelado energético en DesignBuilder: 24 h.
- Diseño del programa en Python: 160 h.
- Exploración de técnicas de Machine Learning: 40 h.
- Redacción de documentos: 60 h.
- Otros: 30h.

Total horas: 324 h.

Total salarios: 6.480 €

Subtotal: 6.839,37 €

Beneficio industrial (10%): 683, 94 €

Total, sin I.V.A: 7.523,31 €

**Total, con I.V.A (21%): 9.103,20 €**