

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

ADAPTACIÓN COMPUTACIONAL DE MODELOS ENERGÉTICOS PARA ESTUDIOS DE INCIDENCIA SOCIAL

Autor: Reglero García, Beatriz.

Director: Romero Mora, José Carlos.

Co-Directora: Morán Camacho, María.

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

Madrid

Junio 2026

Anexo I: Declaración de originalidad

DECLARACIÓN DE ORIGINALIDAD

Declaro bajo mi responsabilidad que el Proyecto presentado con el título **Adaptación Computacional de Modelos Energéticos para Estudios de Incidencia Social** en la ETS de Ingeniería – ICAI de la Universidad Pontificia Comillas en el curso académico **[2025-2026]** es de mi autoría y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente, y la información que ha sido tomada de otros documentos está debidamente referenciada.

USO DE INTELIGENCIA ARTIFICIAL

Declaro bajo mi responsabilidad que:

☐ No he utilizado Inteligencia Artificial en la elaboración del presente documento.

☒ He utilizado Inteligencia Artificial en la elaboración del presente documento y/o del Anexo B siempre en las condiciones permitidas por la Universidad Pontificia Comillas, es decir, aplicando el Nivel 2 de la Escala de Evaluación de Perkins et al. (2024): *"La IA puede utilizarse para actividades previas a la tarea, como la lluvia de ideas, la descripción y la investigación inicial. Este nivel se centra en el uso de la IA para la planificación, las síntesis y la generación de ideas, pero las evaluaciones deben hacer hincapié en la capacidad de desarrollar y refinar estas ideas de forma independiente"*. En concreto, la Inteligencia Artificial ha sido empleada para:

Apoyar la organización y estructuración del contenido del presente Trabajo Fin de Grado, facilitando la ordenación de ideas y la planificación de apartados, sin alterar el contenido técnico ni las conclusiones desarrolladas por la autora y para traducir y adaptar al inglés.

Firmado (alumno): 

Fecha: 12/06/2026

AUTORIZACIÓN PARA LA ENTREGA DEL PROYECTO

El Director del Proyecto	**El co-Director del Proyecto (si aplica)**
Fdo: 	Fdo:
Fecha: 12/06/2026	Fecha: 12/06/2026

Annex I: Declaration of originality

DECLARATION OF ORIGINALITY

I declare under my responsibility that the Project presented with the title **[Computational Adaptation of Energy Models for Social Impact Studies]** at the ICAI School of Engineering of the Comillas Pontifical University in the academic year **[academic year]** is of my authorship and has not been presented previously for other purposes. The Project is not plagiarised from any other, either totally or partially, and the information that has been taken from other documents is duly referenced.

USE OF ARTIFICIAL INTELLIGENCE

I declare under my responsibility that:

☐ I have not used Artificial Intelligence in the preparation of this document.

☒ I have used Artificial Intelligence in the preparation of this document and/or Annex B under the conditions allowed by Comillas Pontifical University, i.e. applying Level 2 of the Perkins et al. (2024) Assessment Scale: *"AI can be used for pre-task activities such as brainstorming, description and initial research. This level focuses on the use of AI for planning, synthesising and generating ideas, but assessments should emphasise the ability to develop and refine these ideas independently"*. Specifically, Artificial Intelligence has been used to:

To support the organization and structuring of the content of this Final Degree Project, facilitating the ordering of ideas and the planning of sections, without altering the technical content or the conclusions developed by the author, and to translate and adapt fragments into English.

Signature (student): 

Date: 12/06/2026

AUTHORISATION FOR PROJECT DELIVERY

Thesis supervisor	**Thesis deputy supervisor (if any)**
Signature:	Signature:
Date:	Date:

Agradecimientos

Agradezco a mis directores, José Carlos Romero Mora y María Morán Camacho, por su orientación y apoyo durante el desarrollo de este trabajo, así como a la Cátedra de Energía y Pobreza por facilitar los recursos y el contexto necesarios para llevar a cabo esta investigación.

ADAPTACIÓN COMPUTACIONAL DE MODELOS ENERGÉTICOS PARA ESTUDIOS DE INCIDENCIA SOCIAL

Autor: Reglero García, Beatriz.
Director: Romero Mora, José Carlos.
Codirectora: Morán Camacho, María.
Entidad Colaboradora: ICAI – Universidad Pontificia Comillas.

RESUMEN DEL PROYECTO

Este Trabajo Fin de Grado describe la migración del modelo de cálculo del gasto energético requerido en hogares españoles, desarrollado por la Cátedra de Energía y Pobreza de la Universidad Pontificia Comillas, desde Engineering Equation Solver (EES) a Python. El modelo, denominado DIAGNÓSTICO, estima la demanda, el consumo y el gasto energético teórico de un hogar a partir de las características de la vivienda, sus instalaciones y el perfil de ocupación del hogar. La nueva versión incorpora una interfaz desarrollada con Streamlit y se valida mediante la comparación de sus resultados con los obtenidos en EES.

Palabras clave: pobreza energética, gasto energético, demanda, consumo, Python, EES y DIAGNÓSTICO.

1. Introducción

El acceso a la energía es necesario para mantener unas condiciones de vida adecuadas en el hogar. En el ámbito residencial, esto incluye usos básicos como la calefacción, la refrigeración, el agua caliente sanitaria (ACS), la iluminación y los electrodomésticos. Cuando un hogar no puede asumir el coste de cubrir estas necesidades, o reduce su consumo por motivos económicos, se encuentra en una situación de pobreza energética [1].

En España, una parte importante del parque residencial fue construida antes de la entrada en vigor de normativas exigentes de eficiencia energética. Esto hace que muchas viviendas necesiten más energía para alcanzar unas condiciones mínimas de confort, especialmente cuando tienen un aislamiento deficiente o sistemas poco eficientes. En los hogares vulnerables, el problema se agrava porque un aumento del precio de la energía puede traducirse en dificultades para mantener una temperatura adecuada o para pagar las facturas.

Además, el gasto energético de una vivienda no depende únicamente del precio de la energía. Factores como la ubicación geográfica, las características constructivas

del edificio, los sistemas instalados o el número de ocupantes influyen de forma significativa en las necesidades energéticas del hogar. Por eso, dos viviendas con niveles de renta similares pueden presentar demandas energéticas muy diferentes.

Por otro lado, el consumo observado no siempre refleja bien las necesidades reales de un hogar. Hay familias que registran un gasto energético reducido no porque requieran menos energía, sino porque sus limitaciones económicas les impiden consumir lo necesario. Esta situación, denominada pobreza energética oculta, hace necesario estimar el consumo y el gasto energético que cada hogar debería asumir para satisfacer unas condiciones básicas de bienestar y confort.

Para estudiar este tipo de situaciones, la Cátedra de Energía y Pobreza de la Universidad Pontificia Comillas desarrolló la herramienta DIAGNÓSTICO, que estima el gasto energético teórico de los hogares españoles combinando características de la vivienda, la localización, los sistemas instalados y el perfil de ocupación. La herramienta original estaba implementada en EES, un entorno útil para el cálculo y la validación del modelo, pero poco flexible para automatizar simulaciones o integrarse con aplicaciones web.

Este Trabajo Fin de Grado parte de esa limitación: trasladar el modelo DIAGNÓSTICO a Python para conservar su lógica de cálculo, pero hacerlo más accesible y operativo para futuros estudios sobre pobreza energética. Además, la nueva estructura permite plantear futuras ampliaciones, como añadir nuevos módulos de cálculo, conectar la herramienta con bases de datos de hogares vulnerables o desplegarla en una aplicación web.

2. Definición del Proyecto

Este Trabajo Fin de Grado tiene como objetivo principal trasladar a Python el modelo DIAGNÓSTICO. La herramienta original estaba implementada en Engineering Equation Solver (EES), un entorno útil para el desarrollo inicial del modelo, pero con limitaciones a la hora de automatizar simulaciones, trabajar con grandes bases de datos o integrarse en una aplicación web.

El proyecto parte del modelo existente y busca reproducir su funcionamiento en un entorno más abierto y flexible, sin cambiar la metodología de base. Para ello, se han trasladado las funciones de cálculo a Python, se han exportado y adaptado las tablas de referencia utilizadas por EES y se ha reorganizado el código en módulos independientes. Esta estructura permite separar el acceso a datos, el motor de cálculo, la gestión del flujo de ejecución y la interfaz de usuario.

Además de la migración del motor de cálculo, se ha desarrollado una interfaz con Streamlit que permite introducir los datos del hogar de forma ordenada y consultar los resultados sin necesidad de trabajar directamente con el código.

En la práctica, esto hace que la herramienta sea más manejable y deja abierta la posibilidad de ampliarla en el futuro. La nueva implementación busca que el

modelo sea más operativo para futuros estudios de pobreza energética, especialmente en escenarios en los que sea necesario ejecutar muchas simulaciones o conectar la herramienta con otras fuentes de datos.

La Figura 5 muestra el orden de cálculo que sigue el modelo original DIAGNÓSTICO.

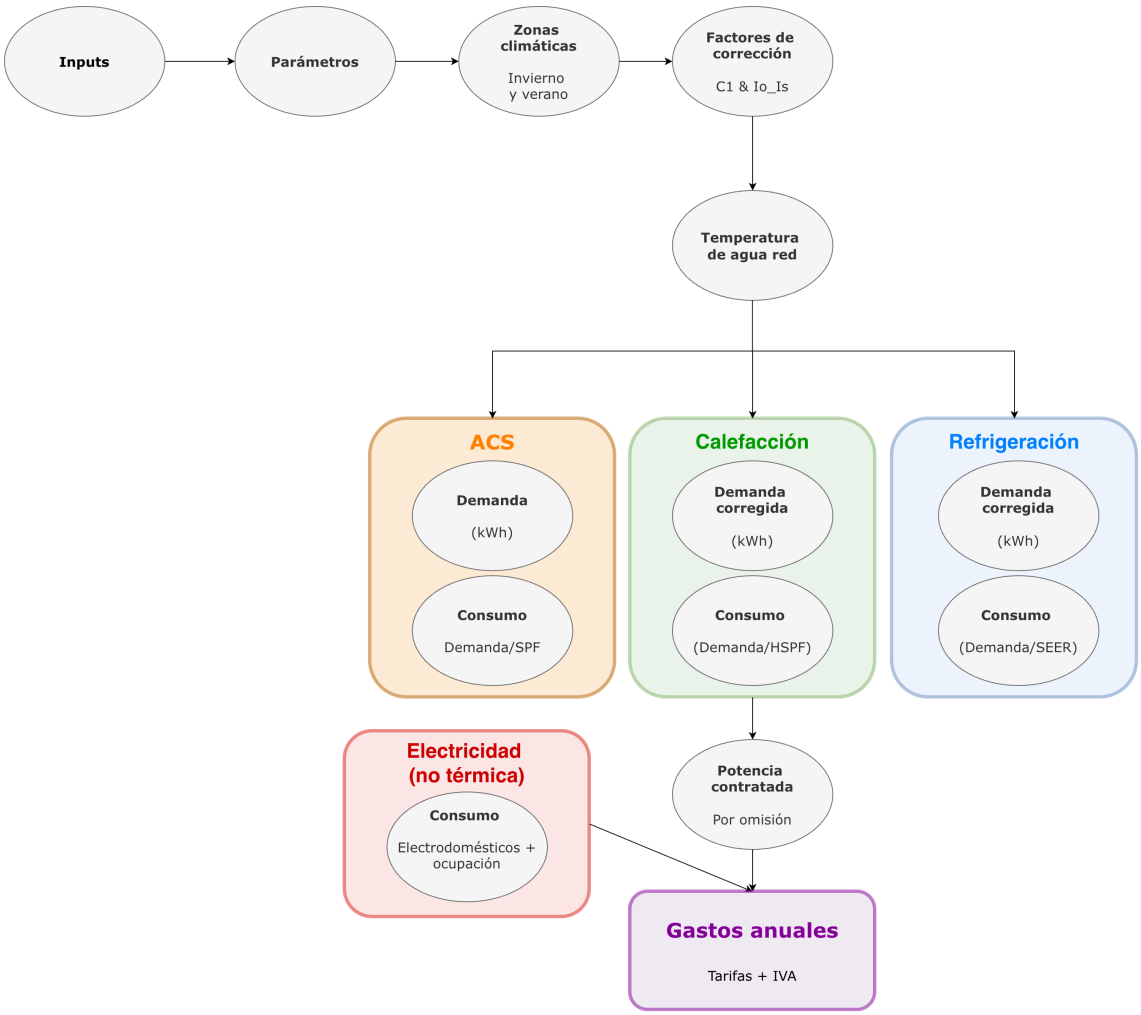


Figura 1: Orden de cálculo del modelo original DIAGNÓSTICO.

Finalmente, la versión en Python se valida mediante una comparación directa con el modelo original en EES. Esta validación permite comprobar si la migración mantiene el comportamiento del modelo original y analizar posibles diferencias debidas al cambio de entorno de cálculo.

3. Descripción del modelo

The original model is based on four reports published by the Chair of Energy and Poverty, specifically: *Caracterización del comportamiento energético en una muestra*

de hogares españoles [2]; *Desarrollo de un modelo de cálculo de gasto eléctrico teórico en los hogares españoles* [3]; *Estudio del gasto térmico teórico en refrigeración para hogares españoles* [4]; and *Investigación aplicada para el desarrollo de una herramienta web de cálculo del gasto eléctrico teórico para hogares españoles* [1].

El modelo DIAGNÓSTICO calcula el gasto energético de un hogar a partir de los datos introducidos por el usuario y de un conjunto de tablas de referencia con parámetros normativos, climáticos y económicos.

Entre las entradas principales se incluyen la provincia y altitud de la vivienda, el tipo de edificio, la superficie útil, el número de ocupantes, la antigüedad o calificación energética, los sistemas de calefacción, refrigeración y ACS, las fuentes de energía utilizadas y el equipamiento eléctrico del hogar.

En cuanto al cálculo energético, el modelo distingue tres niveles: demanda, consumo y gasto. La demanda representa la energía necesaria para cubrir un uso determinado; el consumo representa la energía que debe suministrarse al sistema teniendo en cuenta su rendimiento; y el gasto representa el coste económico asociado a ese consumo. El funcionamiento del modelo se organiza en tres bloques principales: el módulo térmico de calefacción y ACS, el módulo de refrigeración y el módulo eléctrico.

Módulo térmico.

Por un lado, la demanda de calefacción se calcula a partir de la zona climática de la localidad, que se determina combinando la capital de provincia y la altitud. Sobre la demanda de referencia normativa se aplican factores de corrección relacionados con el nivel de aislamiento del edificio y la tipología de la vivienda.

Por otro lado, la demanda de ACS depende del número de ocupantes y de la temperatura mensual del agua de red, corregida por altitud respecto a la capital de provincia. Por último, el consumo se obtiene dividiendo la demanda entre el rendimiento estacional correspondiente, y el gasto se calcula aplicando tarifas, impuestos y términos fijos.

Módulo de refrigeración. Este módulo sigue una estructura parecida a la calefacción, pero aplicada a las condiciones de verano.

La demanda depende de la zona climática de verano, del tipo de vivienda, del nivel de aislamiento y de la superficie climatizada. Las localidades con zona climática de verano igual a 1 tienen demanda de refrigeración nula. El consumo se obtiene dividiendo la demanda entre el SEER del equipo y el gasto se calcula a partir de la tarifa eléctrica correspondiente.

Módulo eléctrico. El módulo eléctrico estima el consumo asociado a usos no térmicos: electrodomésticos, iluminación y dispositivos electrónicos. El modelo distingue entre aparatos de uso variable a lo largo del día, como cocina, lavadora, lavavajillas, televi-

sión u ordenador, y aparatos de consumo más estable, como frigorífico, congelador, microondas, iluminación o dispositivos móviles. Además, incorpora factores de ajuste por ocupación, ya que el tiempo que los miembros del hogar pasan en la vivienda influye en el consumo eléctrico.

Los datos de entrada que requiere el modelo se recogen en la Tabla 5.

Tabla 1: Datos de entrada del modelo DIAGNÓSTICO

Categoría	**Variables**
Localización	Provincia y altitud.
Edificio	Tipo de vivienda, época de construcción o calificación energética y superficie.
Ocupación	Número de ocupantes y situación laboral de cada miembro.
Instalaciones térmicas	Sistema de calefacción, sistema de ACS, tipo de instalación y combustible.
Instalaciones eléctricas	Presencia de electrodomésticos, iluminación y dispositivos electrónicos.
Refrigeración	Porcentaje de superficie climatizada en verano.

El desarrollo de la herramienta en Python se estructuró en cuatro módulos con distintas responsabilidades: `app.py`, encargado de la interfaz de usuario; `help.py`, responsable del acceso a las tablas de referencia; `model.py`, donde se concentra la lógica de cálculo; y `main.py`, que coordina el flujo de ejecución entre la interfaz, las funciones auxiliares y el motor de cálculo. La Figura 6 muestra de forma esquemática cómo se relacionan estos módulos.

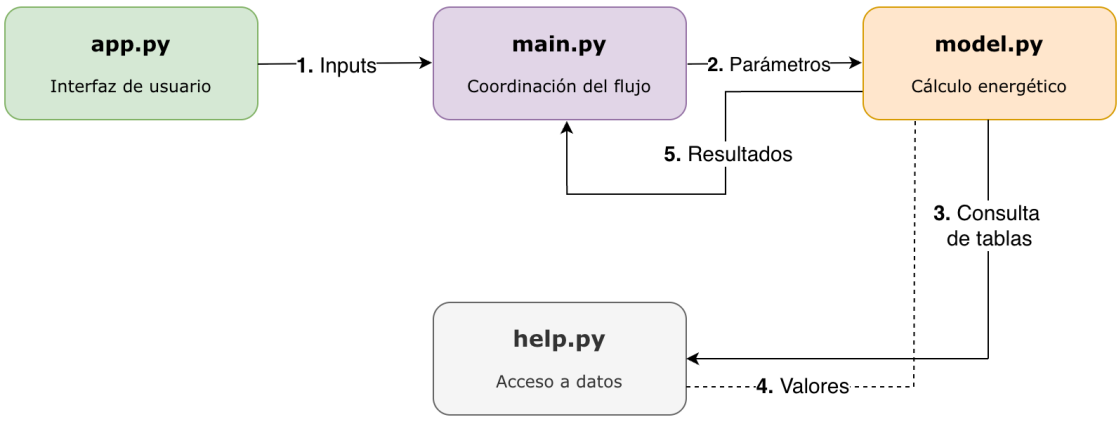


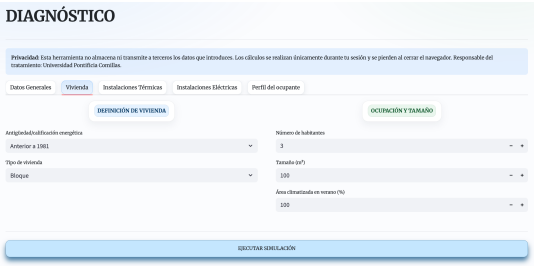
Figura 2: Diagrama de relaciones entre los módulos del sistema.

La interfaz de usuario se estructura en varias pestañas.

La pestaña de datos generales recoge la provincia y la altitud de la localidad. La pestaña de vivienda permite introducir el tipo de vivienda, la superficie, el número de habitantes, la antigüedad o calificación energética y el porcentaje de superficie climatizada en verano. La pestaña de instalaciones térmicas recoge los sistemas de calefacción y ACS, junto con el tipo de instalación. La pestaña de instalaciones eléctricas permite indicar la presencia de los distintos electrodomésticos y dispositivos del hogar. Por último, la pestaña de perfil del ocupante recoge la situación de los miembros del hogar.



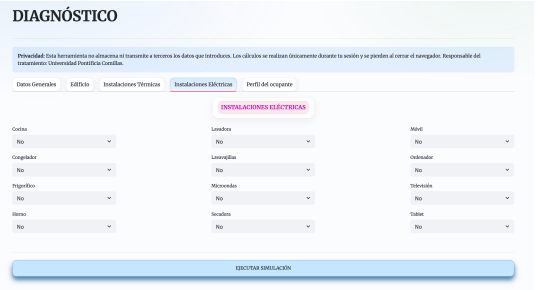
(a) Datos generales.



(b) Vivienda.



(c) Instalaciones térmicas.



(d) Instalaciones eléctricas.

Figura 3: Pestañas principales de entrada de datos de la nueva interfaz en Streamlit.

Una vez introducidos los datos, el usuario puede ejecutar la simulación. La aplicación recoge los valores almacenados, llama al motor de cálculo y muestra los resultados en una página independiente. Esta página presenta la demanda energética anual, el consumo anual y el gasto anual, desglosados por sistema.



Figura 4: Página de resultados finales de la nueva interfaz.

El código desarrollado y las tablas necesarias para ejecutar el modelo se encuentran almacenados en un repositorio privado de GitHub. El acceso al repositorio está disponible en el Anexo II, y el acceso puede solicitarse a través del correo universitario de la autora: `202202350@alu.comillas.edu`.

4. Resultados

Para comprobar que la migración a Python reproduce correctamente el modelo original, se llevó a cabo una comparación sistemática con EES. La validación se basó en 25 casos de prueba diseñados para cubrir al menos una vez las principales opciones de entrada del modelo: sistemas de calefacción, sistemas de ACS, tipos de instalación, viviendas en bloque y unifamiliares, calificaciones energéticas, perfiles de ocupación, provincias de distintas zonas climáticas y diferentes altitudes.

Para cada caso de prueba se introdujeron los mismos datos en EES y en Python. Posteriormente, se compararon 16 variables de salida: demanda, consumo y gasto de calefacción, refrigeración, ACS y electricidad no térmica, además del gasto en combustibles y del gasto eléctrico total con y sin refrigeración.

La precisión del modelo se evaluó mediante el error relativo porcentual medio por variable de salida.

Los resultados se muestran en la Tabla 6.

Tabla 2: Error relativo porcentual medio por variable de salida

Variable	$\overline{\lvert\varepsilon\rvert}$ (%)	$\lvert\varepsilon\rvert_{\text{máx}}$ (%)
Demanda energética (kWh)		
Calefacción	0,0087	0,0358
Refrigeración	0,0053	0,0256
ACS	0,0155	0,0356
Electricidad no térmica	0,0085	0,0192
Consumo (kWh)		
Calefacción	0,0070	0,0319
ACS	0,0122	0,0299
Refrigeración	0,0254	0,0953
Electricidad no térmica	0,0085	0,0192
Gasto (€)		
Calefacción	0,0063	0,0381
ACS	0,0023	0,0165
Refrigeración	0,0001	0,0030
Combustibles	0,0032	0,0202
Electricidad no térmica (sin ref.)	0,0050	0,0298
Electricidad no térmica (con ref.)	0,0054	0,0298
Electricidad total (sin ref.)	0,0048	0,0298
Electricidad total (con ref.)	0,0048	0,0298

En todos los casos, los errores medios quedan por debajo del 0,03 %, y los máximos individuales no superan el 0,10 %. La variable con mayor error medio es el consumo de refrigeración, con un 0,0254 %, seguida de la demanda de ACS, con un 0,0155 %.

A nivel de categorías, se calculó el error absoluto medio para las tres categorías principales del modelo: demanda, consumo y gasto. Como se observa en la Tabla 7, el MAE más alto corresponde al consumo, con un 0,0133 %, seguido de la demanda, con un 0,0095 %, y del gasto, con un 0,0049 %.

Tabla 3: Error absoluto medio agregado por categoría

Categoría	**MAE (%)**
Demanda energética	0,0095
Consumo	0,0133
Gasto	0,0049

Esta jerarquía de errores responde a la propia estructura del modelo. El consumo se obtiene a partir de la demanda y del rendimiento estacional del sistema, por lo que pequeñas diferencias de redondeo pueden trasladarse al paso siguiente. El gasto, en cambio, resulta de aplicar tarifas comunes a ambos modelos, lo que tiende a reducir las diferencias relativas acumuladas.

También se analizó si el error dependía de la zona climática de verano. Para ello, se estudió el consumo de refrigeración, que es la variable con mayor error medio del conjunto. Los resultados se resumen en la Tabla 8.

Tabla 4: Error relativo medio del consumo de refrigeración por zona climática de verano

Zona climática de verano	**Error medio (%)**
1	0,0000
2	0,0603
3	0,0370
4	0,0226

La zona 1 presenta error cero porque el modelo asigna demanda de refrigeración nula en estas localidades. En el resto de zonas no se observa una relación directa entre la severidad climática de verano y el error obtenido. La zona 2 presenta el mayor error medio, mientras que la zona 4, que corresponde a condiciones de verano más exigentes y a mayores demandas de refrigeración, presenta el menor error medio entre las zonas con refrigeración.

Esto permite concluir que las diferencias entre ambos modelos no aumentan con la magnitud de la demanda ni con la intensidad de la zona climática. Por tanto, los errores observados no parecen estar asociados a un comportamiento estructural del

modelo en determinadas condiciones climáticas, sino a pequeñas diferencias numéricas derivadas del cambio de entorno de cálculo. Este resultado refuerza la validez de la implementación en Python, ya que la migración mantiene el comportamiento del modelo original con márgenes de error despreciables.

En conjunto, los resultados no muestran ningún patrón sistemático asociado a la provincia, al sistema térmico, a la calificación energética o a la zona climática. Esto refuerza la conclusión de que las diferencias entre ambos modelos son atribuibles a redondeos numéricos internos y no a errores de lógica en la implementación.

5. Conclusiones

Las conclusiones obtenidas en este Trabajo Fin de Grado pueden agruparse en tres aspectos: la viabilidad de la migración, la utilidad de la nueva arquitectura y la precisión de los resultados.

En primer lugar, se concluye que la migración del modelo DIAGNÓSTICO de EES a Python es viable sin necesidad de modificar la lógica de cálculo original. La estrategia de traducción función a función permitió mantener una correspondencia directa entre ambos entornos y verificar de forma progresiva que el comportamiento de cada bloque era el esperado.

En segundo lugar, la organización modular del código ha resultado adecuada para una herramienta de estas características. Separar la interfaz, el acceso a datos, el motor de cálculo y el flujo principal facilita la depuración y el mantenimiento del sistema. Además, permite que futuras modificaciones puedan realizarse de forma localizada, por ejemplo, actualizando una tarifa, añadiendo una variable de entrada o incorporando un nuevo módulo de cálculo.

En tercer lugar, la interfaz desarrollada en Streamlit mejora el uso práctico del modelo respecto a la implementación original en EES. El usuario puede introducir los datos de forma guiada, ejecutar la simulación y consultar los resultados de manera ordenada. Además, la aplicación incorpora comprobaciones que permiten evitar incoherencias en los datos de entrada, como la falta de correspondencia entre el número total de habitantes y el perfil de ocupación declarado.

Desde el punto de vista numérico, la validación confirma que el modelo en Python reproduce correctamente el comportamiento del modelo original. Las diferencias observadas son muy reducidas, se sitúan por debajo del 0,03 % de error medio en todas las variables y no presentan patrones sistemáticos. Por tanto, se consideran atribuibles a redondeos internos y no a errores de implementación.

En definitiva, el trabajo permite disponer de una versión funcional, modular y validada del modelo DIAGNÓSTICO en Python. La nueva implementación mantiene la lógica del modelo original y facilita su uso en análisis más amplios, especialmente en escenarios donde sea necesario ejecutar muchas simulaciones o trabajar con bases de datos de hogares.

6. Referencias

[1] Arenas Pinilla, E., Barrella, R., Linares Hurtado, J. I. y Romero Mora, J. C. (2021). *Investigación aplicada para el desarrollo de una herramienta web de cálculo del gasto eléctrico teórico para hogares españoles*. Cátedra de Energía y Pobreza, Universidad Pontificia Comillas.

[2] Arenas Pinilla, E., Barrella, R., Linares Hurtado, J. I., Romero Mora, J. C., Foronda Díez, C., Pesque Castillo, C. y Díez Alzueta, L. (2019). *Caracterización del comportamiento energético en una muestra de hogares españoles*. ECODES; Cátedra de Energía y Pobreza.

[3] Arenas Pinilla, E., Barrella, R., Linares Hurtado, J. I. y Romero Mora, J. C. (2020). *Desarrollo de un modelo de cálculo de gasto eléctrico teórico en los hogares españoles*. Cátedra de Energía y Pobreza, Universidad Pontificia Comillas.

[4] Barrella, R., Arenas Pinilla, E., Linares Hurtado, J. I. y Romero Mora, J. C. (2022). *Estudio del gasto térmico teórico en refrigeración para hogares españoles*. Cátedra de Energía y Pobreza, Universidad Pontificia Comillas.

COMPUTATIONAL ADAPTATION OF ENERGY MODELS FOR SOCIAL IMPACT STUDIES

Author: Reglero García, Beatriz.
Supervisor: Romero Mora, José Carlos.
Co-supervisor: Morán Camacho, María.
Collaborating Entity: ICAI – Universidad Pontificia Comillas.

PROJECT ABSTRACT

This Final Degree Project describes the migration of a model for calculating the required energy expenditure of Spanish households, developed by the Chair of Energy and Poverty at Universidad Pontificia Comillas, from Engineering Equation Solver (EES) to Python. The model, named DIAGNÓSTICO, estimates the theoretical energy demand, consumption and expenditure of a household based on the characteristics of the dwelling, its installations and the household occupancy profile. The new version incorporates an interface developed with Streamlit and is validated by comparing its results with those obtained in EES.

Keywords: energy poverty, energy expenditure, demand, consumption, Python, EES and DIAGNÓSTICO.

1. Introduction

Access to energy is necessary to maintain adequate living conditions in the home. In the residential sector, this includes basic uses such as heating, cooling, domestic hot water (DHW), lighting and household appliances. When a household cannot afford the cost of meeting these needs, or reduces its consumption for economic reasons, it is in a situation of energy poverty [1].

In Spain, a significant share of the residential building stock was built before demanding energy efficiency regulations came into force. As a result, many dwellings require more energy to reach minimum comfort conditions, especially when they have poor insulation or inefficient systems. In vulnerable households, the problem becomes more severe because an increase in energy prices can lead to difficulties in maintaining an adequate indoor temperature or paying energy bills.

In addition, the energy expenditure of a dwelling does not depend only on the price of energy. Factors such as geographical location, the construction characteristics of the building, the systems installed or the number of occupants have a significant influence on the energy needs of the household. For this reason, two dwellings with

similar income levels may present very different energy demands.

On the other hand, observed consumption does not always accurately reflect the real needs of a household. Some families record low energy expenditure not because they require less energy, but because their economic constraints prevent them from consuming the amount needed. This situation, known as hidden energy poverty, makes it necessary to estimate the energy consumption and expenditure that each household should assume in order to meet basic conditions of well-being and comfort.

To study these situations, the Chair of Energy and Poverty at Universidad Pontificia Comillas developed the DIAGNÓSTICO tool, which estimates the theoretical energy expenditure of Spanish households by combining dwelling characteristics, location, installed systems and occupancy profile. The original tool was implemented in EES, an environment that is useful for calculation and model validation, but not very flexible for automating simulations or integrating the model into web applications.

This Final Degree Project starts from that limitation: migrating the DIAGNÓSTICO model to Python in order to preserve its calculation logic, while making it more accessible and operational for future studies on energy poverty. In addition, the new structure makes it possible to consider future extensions, such as adding new calculation modules, connecting the tool to databases of vulnerable households or deploying it as a web application.

2. Project Definition

The main objective of this Final Degree Project is to migrate the DIAGNÓSTICO model to Python. The original tool was implemented in Engineering Equation Solver (EES), an environment that was useful for the initial development of the model, but which presents limitations when it comes to automating simulations, working with large databases or integrating the model into a web application.

The project starts from the existing model and seeks to reproduce its behaviour in a more open and flexible environment, without changing the underlying methodology. To this end, the calculation functions have been transferred to Python, the reference tables used by EES have been exported and adapted, and the code has been reorganised into independent modules. This structure makes it possible to separate data access, the calculation engine, the execution flow management and the user interface.

In addition to migrating the calculation engine, an interface has been developed with Streamlit, allowing the user to enter household data in an organised way and consult the results without having to work directly with the code.

In practice, this makes the tool easier to use and leaves open the possibility of extending it in the future. The new implementation aims to make the model more operational for future energy poverty studies, especially in scenarios where many simulations need to be run or the tool needs to be connected to other data sources.

Figure 5 shows the calculation order followed by the original DIAGNÓSTICO model.

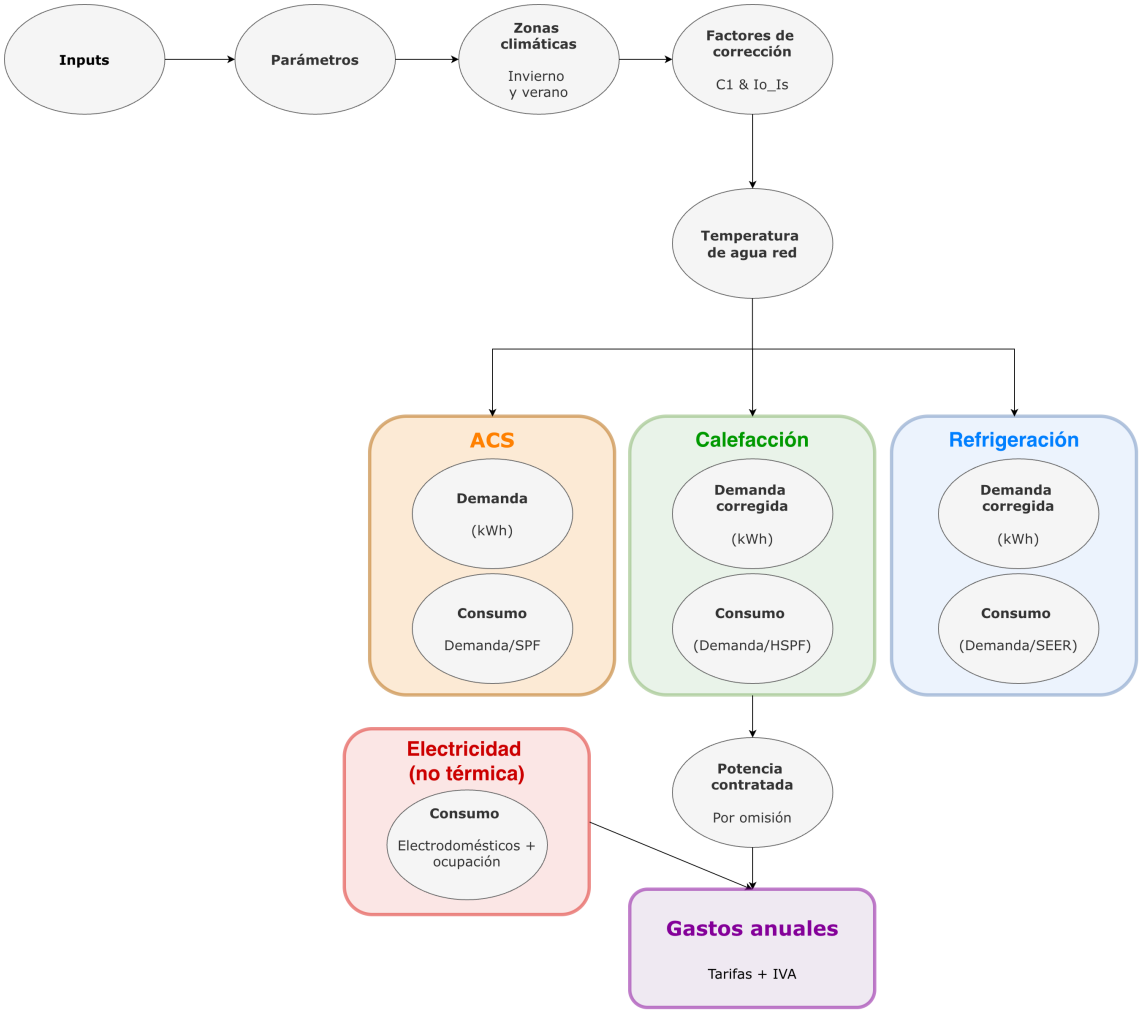


Figura 5: Calculation order of the original DIAGNÓSTICO model.

Finally, the Python version is validated through a direct comparison with the original model in EES. This validation makes it possible to verify whether the migration preserves the behaviour of the original model and to analyse possible differences resulting from the change in calculation environment.

3. Model Description

The DIAGNÓSTICO model calculates the energy expenditure of a household based on the data entered by the user and a set of reference tables containing regulatory, climatic and economic parameters.

The main inputs include the province and altitude of the dwelling, the type of building, the useful floor area, the number of occupants, the age or energy rating of

the dwelling, the heating, cooling and DHW systems, the energy sources used and the household electrical equipment.

In terms of energy calculation, the model distinguishes between three levels: demand, consumption and expenditure. Demand represents the energy required to cover a specific use; consumption represents the energy that must be supplied to the system, taking into account its efficiency; and expenditure represents the economic cost associated with that consumption. The model is organised into three main calculation blocks: the thermal module for heating and DHW, the cooling module and the electrical module.

Thermal module.

On the one hand, heating demand is calculated from the climatic zone of the locality, which is determined by combining the provincial capital and the altitude. Correction factors related to the insulation level of the building and the dwelling typology are then applied to the regulatory reference demand.

On the other hand, DHW demand depends on the number of occupants and the monthly temperature of the mains water, corrected by altitude with respect to the provincial capital. Finally, consumption is obtained by dividing demand by the corresponding seasonal performance factor, and expenditure is calculated by applying tariffs, taxes and fixed charges.

Cooling module. This module follows a structure similar to that of heating, but applied to summer conditions.

Demand depends on the summer climatic zone, the type of dwelling, the insulation level and the cooled floor area. Localities with summer climatic zone equal to 1 have zero cooling demand. Consumption is obtained by dividing demand by the SEER of the equipment, and expenditure is calculated based on the corresponding electricity tariff.

Electrical module. The electrical module estimates consumption associated with non-thermal uses: household appliances, lighting and electronic devices. The model distinguishes between appliances whose use varies throughout the day, such as cooking appliances, washing machines, dishwashers, televisions or computers, and appliances with more stable consumption, such as refrigerators, freezers, microwaves, lighting or mobile devices. It also incorporates occupancy adjustment factors, since the time household members spend at home influences electricity consumption.

The input data required by the model are summarised in Table 5.

Tabla 5: Input data of the DIAGNÓSTICO model

Category	**Variables**
Location	Province and altitude.
Building	Type of dwelling, construction period or energy rating, and floor area.
Occupancy	Number of occupants and employment status of each household member.
Thermal installations	Heating system, DHW system, type of installation and fuel.
Electrical installations	Presence of household appliances, lighting and electronic devices.
Cooling	Percentage of cooled floor area in summer.

The development of the tool in Python was structured into four modules with different responsibilities: `app.py`, responsible for the user interface; `help.py`, responsible for accessing the reference tables; `model.py`, where the calculation logic is concentrated; and `main.py`, which coordinates the execution flow between the interface, the auxiliary functions and the calculation engine. Figure 6 schematically shows how these modules are related.

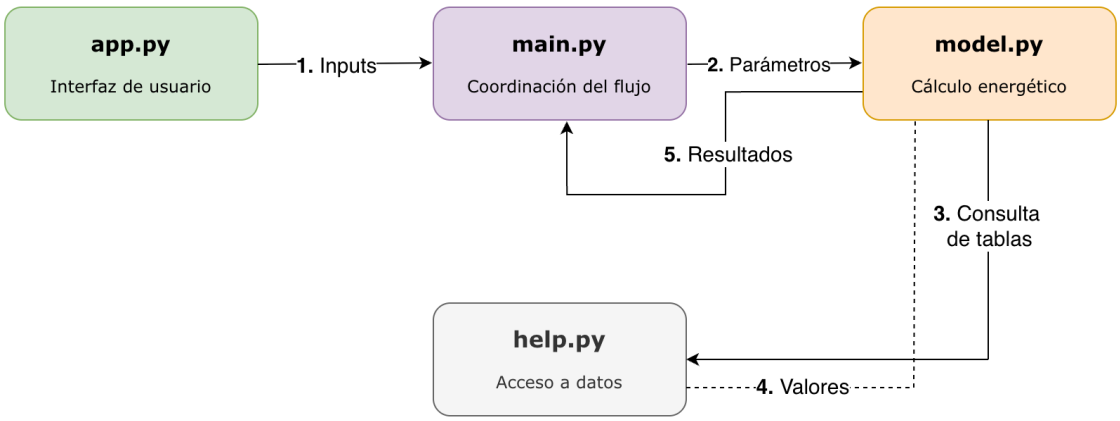
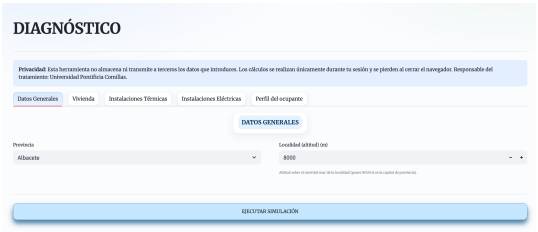


Figura 6: Relationship diagram between the system modules.

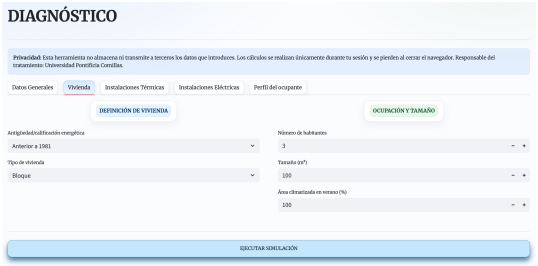
The user interface is structured into several tabs.

The general data tab collects the province and altitude of the locality. The dwelling tab allows the user to enter the type of dwelling, floor area, number of inhabitants, construction period or energy rating, and the percentage of cooled floor area in summer. The thermal installations tab collects the heating and DHW systems,

together with the type of installation. The electrical installations tab allows the user to indicate the presence of the different household appliances and devices. Finally, the occupant profile tab collects the status of the household members.



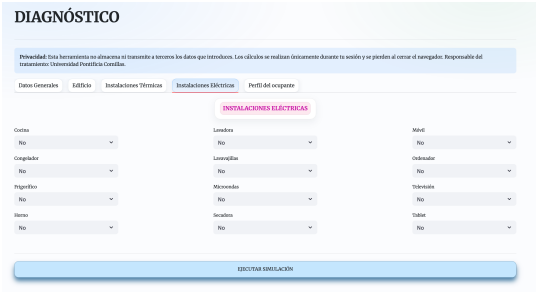
(a) General data.



(b) Dwelling.



(c) Thermal installations.



(d) Electrical installations.

Figura 7: Main data entry tabs of the new Streamlit interface.

Once the data have been entered, the user can run the simulation. The application collects the stored values, calls the calculation engine and displays the results on a separate page. This page presents the annual energy demand, annual consumption and annual expenditure, broken down by system.

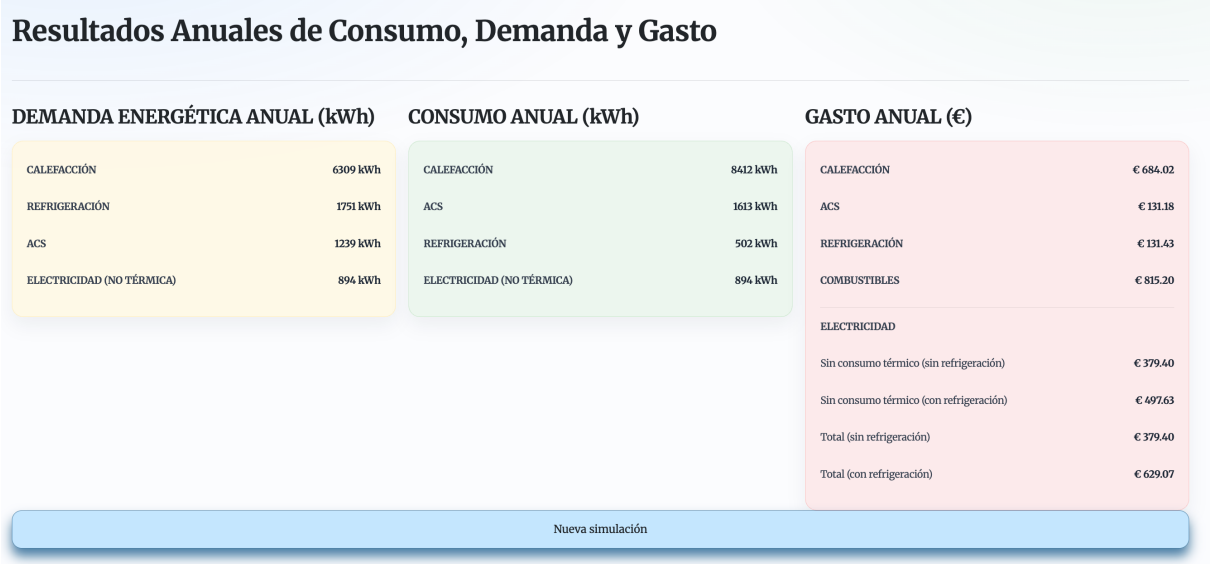


Figura 8: Final results page of the new interface.

The code developed and the tables required to run the model are stored in a private GitHub repository. The repository is referenced in Annex 5.4, and access can be requested through the author's university email address: `202202350@alu.comillas.edu`.

4. Results

To verify that the migration to Python correctly reproduces the original model, a systematic comparison with EES was carried out. The validation was based on 25 test cases designed to cover at least once the main input options of the model: heating systems, DHW systems, types of installation, dwellings in blocks and single-family houses, energy ratings, occupancy profiles, provinces from different climatic zones and different altitudes.

For each test case, the same input data were entered in EES and in Python. Subsequently, 16 output variables were compared: demand, consumption and expenditure for heating, cooling, DHW and non-thermal electricity, as well as fuel expenditure and total electricity expenditure with and without cooling.

The accuracy of the model was evaluated using the mean relative percentage error for each output variable.

The results are shown in Table 6.

Tabla 6: Mean relative percentage error by output variable

Variable	$\overline{\lvert\varepsilon\rvert}$ **(%)**	$\lvert\varepsilon\rvert_{\text{máx}}$ **(%)**
Energy demand (kWh)		
Heating	0,0087	0,0358
Cooling	0,0053	0,0256
DHW	0,0155	0,0356
Non-thermal electricity	0,0085	0,0192
Consumption (kWh)		
Heating	0,0070	0,0319
DHW	0,0122	0,0299
Cooling	0,0254	0,0953
Non-thermal electricity	0,0085	0,0192
Expenditure (€)		
Heating	0,0063	0,0381
DHW	0,0023	0,0165
Cooling	0,0001	0,0030
Fuels	0,0032	0,0202
Non-thermal electricity (without cooling)	0,0050	0,0298
Non-thermal electricity (with cooling)	0,0054	0,0298
Total electricity (without cooling)	0,0048	0,0298
Total electricity (with cooling)	0,0048	0,0298

In all cases, the mean errors remain below 0,03 %, and the maximum individual errors do not exceed 0,10 %. The variable with the highest mean error is cooling consumption, with 0,0254 %, followed by DHW demand, with 0,0155 %.

At the category level, the mean absolute error was calculated for the three main categories of the model: demand, consumption and expenditure. As shown in Table 7, the highest MAE corresponds to consumption, with 0,0133 %, followed by demand, with 0,0095 %, and expenditure, with 0,0049 %.

Tabla 7: Aggregate mean absolute error by category

Category	**MAE (%)**
Energy demand	0,0095
Consumption	0,0133
Expenditure	0,0049

This hierarchy of errors is explained by the structure of the model itself. Consumption is obtained from demand and the seasonal performance of the system, so small rounding differences may be carried over to the next step. Expenditure, on the other hand, results from applying tariffs that are common to both models, which tends to reduce the accumulated relative differences.

It was also analysed whether the error depended on the summer climatic zone. To do so, cooling consumption was studied, since it is the variable with the highest mean error in the whole set. The results are summarised in Table 8.

Tabla 8: Mean relative error of cooling consumption by summer climatic zone

Summer climatic zone	**Mean error (%)**
1	0,0000
2	0,0603
3	0,0370
4	0,0226

Zone 1 presents zero error because the model assigns zero cooling demand to these localities. In the remaining zones, no direct relationship is observed between the severity of the summer climatic zone and the error obtained. Zone 2 presents the highest mean error, while zone 4, which corresponds to more demanding summer conditions and higher cooling demands, presents the lowest mean error among the zones with cooling demand.

This allows the conclusion that the differences between both models do not increase with the magnitude of demand or with the intensity of the climatic zone. Therefore, the observed errors do not appear to be associated with a structural behaviour of the model under certain climatic conditions, but rather with small numerical differences resulting from the change in calculation environment. This result reinforces the validity of the Python implementation, since the migration preserves the behaviour of the original model with negligible error margins.

Overall, the results do not show any systematic pattern associated with the province, the thermal system, the energy rating or the climatic zone. This reinforces the conclusion that the differences between both models are attributable to internal numerical rounding and not to logical errors in the implementation.

5. Conclusions

The conclusions obtained in this Final Degree Project can be grouped into three aspects: the feasibility of the migration, the usefulness of the new architecture and the accuracy of the results.

Firstly, it can be concluded that the migration of the DIAGNÓSTICO model from EES to Python is feasible without modifying the original calculation logic. The function-by-function translation strategy made it possible to maintain a direct correspondence between both environments and to progressively verify that the behaviour of each block was as expected.

Secondly, the modular organisation of the code has proved suitable for a tool of this kind. Separating the interface, data access, calculation engine and main

execution flow facilitates debugging and system maintenance. In addition, it allows future modifications to be made locally, for example by updating a tariff, adding an input variable or incorporating a new calculation module.

Thirdly, the interface developed in Streamlit improves the practical use of the model compared with the original implementation in EES. The user can enter the data in a guided way, run the simulation and consult the results in an organised manner. In addition, the application incorporates checks that prevent inconsistencies in the input data, such as a mismatch between the total number of inhabitants and the declared occupancy profile.

From a numerical point of view, the validation confirms that the Python model correctly reproduces the behaviour of the original model. The observed differences are very small, remain below 0,03 % mean error for all variables and do not present systematic patterns. Therefore, they are considered to be attributable to internal rounding and not to implementation errors.

In conclusion, this work provides a functional, modular and validated Python version of the DIAGNÓSTICO model. The new implementation preserves the logic of the original model and facilitates its use in broader analyses, especially in scenarios where many simulations need to be run or household databases need to be processed.

6. References

[1] Arenas Pinilla, E., Barrella, R., Linares Hurtado, J. I. y Romero Mora, J. C. (2021). *Investigación aplicada para el desarrollo de una herramienta web de cálculo del gasto eléctrico teórico para hogares españoles*. Cátedra de Energía y Pobreza, Universidad Pontificia Comillas.

[2] Arenas Pinilla, E., Barrella, R., Linares Hurtado, J. I., Romero Mora, J. C., Foronda Díez, C., Pesque Castillo, C. y Díez Alzueta, L. (2019). *Caracterización del comportamiento energético en una muestra de hogares españoles*. ECODES; Cátedra de Energía y Pobreza.

[3] Arenas Pinilla, E., Barrella, R., Linares Hurtado, J. I. y Romero Mora, J. C. (2020). *Desarrollo de un modelo de cálculo de gasto eléctrico teórico en los hogares españoles*. Cátedra de Energía y Pobreza, Universidad Pontificia Comillas.

[4] Barrella, R., Arenas Pinilla, E., Linares Hurtado, J. I. y Romero Mora, J. C. (2022). *Estudio del gasto térmico teórico en refrigeración para hogares españoles*. Cátedra de Energía y Pobreza, Universidad Pontificia Comillas.

Índice de la memoria

Anexo I: Declaración de originalidad I

Annex I: Declaration of originality III

Resumen VI

Abstract XVI

Capítulo 1. Introducción 1
1.1 Motivación del proyecto 2
1.2 Objetivos del proyecto 5

Capítulo 2. Estado de la Técnica 6
2.1 Modelos de gasto energético en hogares 6
2.1.1 Gasto térmico teórico y gasto requerido 7
2.1.2 Gasto eléctrico teórico 7
2.2 Herramientas de modelado energético 8
2.3 Estructura de la herramienta DIAGNÓSTICO 8
2.3.1 Módulo térmico: calefacción y ACS 10
2.3.2 Módulo de refrigeración 13
2.3.3 Módulo eléctrico 14
2.3.4 Salidas del modelo 16
2.3.5 Datos de entrada y tablas de referencia 18
2.3.6 Flujo de cálculo 20

Capítulo 3. Desarrollo Experimental 21
3.1 Planteamiento 21
3.2 Estructura 22
3.3 Metodología 23
3.3.1 Interfaz de usuario 23
3.3.2 Acceso a datos 28
3.3.3 Lógica del cálculo energético 29
3.3.4 Coordinación del flujo 31
3.4 Auditoría de seguridad 33
3.4.1 Dependencias y configuración 33
3.4.2 Control de errores 33

3.4.3 Navegación y privacidad . . . 34

Capítulo 4. Comparación de modelos y análisis de resultados . . . **35**

4.1 Metodología de validación . . . **35**

4.2 Resultados de la validación . . . **37**

4.2.1 Precisión por variable . . . 37

4.2.2 Precisión agregada por categoría . . . 38

4.3 Análisis de resultados . . . **38**

4.3.1 Interpretación del error residual . . . 38

4.3.2 Distribución de errores por zona climática . . . 39

4.3.3 Patrones en el gasto eléctrico . . . 40

4.3.4 Provincias con mayor error por variable . . . 41

4.3.5 Conclusión . . . 42

Capítulo 5. Conclusiones y futuros estudios . . . **43**

5.1 Conclusiones sobre la metodología . . . **43**

5.2 Limitaciones . . . **44**

5.3 Conclusiones sobre los resultados . . . **46**

5.4 Recomendaciones para futuros estudios . . . **47**

Bibliografía . . . **49**

Anexo II: Repositorio del proyecto . . . **51**

Anexo III: Tablas de validación . . . **52**

Anexo IV: Alineación con los Objetivos de Desarrollo Sostenible . . . **55**

.1 ODS 1: Fin de la pobreza . . . **55**

.2 ODS 7: Energía asequible y no contaminante . . . **55**

.3 ODS 10: Reducción de las desigualdades . . . **56**

.4 ODS 11: Ciudades y comunidades sostenibles . . . **56**

Índice de figuras

1 Orden de cálculo del modelo original DIAGNÓSTICO. VIII
2 Diagrama de relaciones entre los módulos del sistema. X
3 Pestañas principales de entrada de datos de la nueva interfaz en Streamlit. XI
4 Página de resultados finales de la nueva interfaz. XI
5 Calculation order of the original DIAGNÓSTICO model. XVIII
6 Relationship diagram between the system modules. XX
7 Main data entry tabs of the new Streamlit interface. XXI
8 Final results page of the new interface. XXII

2.1 Pantalla principal de DIAGNÓSTICO en EES, con los campos de entrada a la izquierda y las salidas a la derecha. 9
2.2 Pestañas de entrada de datos de la herramienta DIAGNÓSTICO en EES. 9
2.3 Secuencia de cálculo de la herramienta DIAGNÓSTICO. 20

3.1 Pestaña de Datos Generales. 23
3.2 Pestaña de Vivienda. 24
3.3 Pestaña de Instalaciones Térmicas. 24
3.4 Pestaña de Instalaciones Eléctricas. 25
3.5 Pestaña de Perfil del ocupante. 25
3.6 Pestaña de resultados finales. 26
3.7 Diagrama de relaciones entre los módulos del sistema. 32

1 Casos de validación: datos de entrada completos. 52
2 Resultados de demanda, consumo y gasto obtenidos por Python y EES para los 25 casos de validación. 52
3 Errores relativos porcentuales y MAE por variable y caso para demanda. 53
4 Errores relativos porcentuales y MAE por variable y caso para consumo. 53
5 Errores relativos porcentuales y MAE por variable y caso para gasto. 54
6 Errores relativos porcentuales por zona climática de verano. 54

Índice de tablas

1 Datos de entrada del modelo DIAGNÓSTICO x
2 Error relativo porcentual medio por variable de salida xii
3 Error absoluto medio agregado por categoría xiii
4 Error relativo medio del consumo de refrigeración por zona climática de verano xiii
5 Input data of the DIAGNÓSTICO model xx
6 Mean relative percentage error by output variable xxiii
7 Aggregate mean absolute error by category xxiii
8 Mean relative error of cooling consumption by summer climatic zone xxiv

4.1 Casos de validación empleados en la comparación Python–EES 36
4.2 Error relativo porcentual medio por variable de salida 37
4.3 Error absoluto medio (MAE) agregado por categoría 38
4.4 Zona climática de invierno y verano de los 25 casos de validación . . . 39
4.5 Error relativo medio del consumo de refrigeración por zona climática de verano 40
4.6 Error relativo máximo individual por variable de salida 41

Capítulo 1. Introducción

El acceso a la energía es uno de los pilares fundamentales del bienestar humano, cuya disponibilidad, o bien falta de ella, condiciona de forma directa las condiciones de vida de las personas. En España, el sector residencial representa entre el 25 % y el 30 % del consumo total de energía final según el IDAE, lo que le da un peso relevante tanto en el balance energético del país como en la economía de las familias.

Los precios de la energía llevan años siendo inestables, y la crisis de 2021-2022 los disparó a niveles que no se habían visto antes en Europa. Detrás de esa subida hubo varios factores: la recuperación económica tras la pandemia, la reducción de reservas de gas y distintos conflictos geopolíticos que afectaron al suministro. A esto se suman tensiones más recientes, como las relacionadas con el estrecho de Ormuz, una de las principales rutas de transporte de petróleo y gas, que siguen generando incertidumbre en los mercados energéticos y contribuyen a nuevas subidas de precio. El problema se agrava además por el estado del parque residencial español: más de la mitad de los edificios en España se construyeron antes de que existiera ninguna normativa de eficiencia energética [5], lo que significa que una gran parte de los hogares parte ya en desventaja frente a cualquier subida de precios.

Tanto el Pacto Verde Europeo como la Agenda 2030 de Naciones Unidas marcan como objetivo garantizar el acceso a una energía asequible y sostenible para todos, sin embargo, reconocen que sigue siendo un problema sin resolver para una parte importante de la población.

En este contexto, la pobreza energética hace referencia a la situación de aquellos hogares que no pueden cubrir adecuadamente sus necesidades básicas de energía (calefacción, refrigeración, agua caliente o electricidad), debido principalmente a unos ingresos insuficientes. Esta situación suele agravarse en viviendas con baja eficiencia energética, donde mantener unas condiciones mínimas de confort requiere un mayor consumo. En la práctica, puede manifestarse de distintas maneras: dificultades para mantener una temperatura adecuada en el hogar, retrasos en el pago de facturas o un gasto energético demasiado elevado en comparación con los ingresos disponibles [6].

La pobreza energética se ha convertido en uno de los problemas sociales más serios de los países desarrollados y las cifras nacionales no reflejan bien la situación de los hogares más vulnerables. Según el Informe de Indicadores de Pobreza Energética en España 2024 elaborado por la Cátedra de Energía y Pobreza de la Universidad Pontificia Comillas [7], el 16,51 % de los hogares españoles presentaba gasto energético desproporcionado en 2024, el 17,56 % de la población declaró no poder mantener una temperatura adecuada en invierno, y cerca del 10 % acumulaba retrasos en el pago de sus facturas energéticas. Además, el 27,10 % de los hogares gastó en energía

menos de lo necesario para cubrir sus necesidades térmicas y eléctricas, situándose en situación de pobreza energética oculta.

Aunque la moderación de los precios de la energía en 2024 permitió una ligera mejora en algunos indicadores, como el gasto desproporcionado (2M) o la temperatura inadecuada en invierno, otros continúan empeorando, como el gasto insuficiente (M/2), que alcanzó su máximo histórico en 2024 con un 12,87 % de los hogares, o los retrasos en el pago de facturas. Además, aunque buena parte del debate se ha centrado en la calefacción y la refrigeración, el consumo eléctrico vinculado a electrodomésticos, iluminación y dispositivos también pesa considerablemente en la economía de los hogares con menos ingresos.

El programa «Ni un hogar sin energía» ha recogido datos de cerca de 4.000 hogares en situación de vulnerabilidad económica. Los análisis realizados por la Cátedra de Energía y Pobreza de la Universidad Pontificia Comillas sobre esa muestra son muy reveladores: más del 40 % de esos hogares declaraban no poder mantener su casa a una temperatura de confort en invierno, frente al 8 % de la población general [2].

Estos datos ponen de manifiesto que las cifras tienden a subestimar la magnitud real del problema y que para diseñar ayudas que funcionen hacen falta modelos que combinen las características de la vivienda: superficie, zona climática, tipo de instalaciones, con las del hogar: cuántas personas viven en ella, cómo está compuesta la familia. En el ámbito térmico ya existe una metodología basada en el Código Técnico de la Edificación que permite estimar el gasto térmico teórico (GTT) y el gasto térmico requerido (GTR). Para la parte eléctrica, la Cátedra de Energía y Pobreza desarrolló en 2020, en colaboración con ECODES, un modelo de cálculo del gasto eléctrico teórico (GELT) [3].

Al aplicarlo a la base de datos de «Ni un hogar sin energía», se observó que el consumo real de estos hogares era entre un 16 % y un 20 % inferior al teórico, según los distintos modelos aplicados, y que entre el 61 % y el 65 % de los hogares visitados consumían menos electricidad de la que necesitarían [1]. Se trata de lo que se conoce como pobreza energética oculta: hogares que consumen menos energía de la necesaria por incapacidad económica para afrontar ese gasto, y que por tanto no aparecen en las estadísticas convencionales como casos graves [8]. Estos resultados apuntan a una brecha real entre lo que estos hogares consumen y lo que necesitarían consumir para vivir en condiciones adecuadas, lo que refuerza la necesidad de contar con herramientas fiables para estimar sus necesidades energéticas reales.

1.1. MOTIVACIÓN DEL PROYECTO

Para dar respuesta a la necesidad de estimar el gasto energético de los hogares vulnerables, la Cátedra de Energía y Pobreza ha desarrollado a lo largo de los últimos años una metodología de cálculo que combina dos componentes principales. Por un lado, el módulo térmico, basado en el Código Técnico de la Edificación (CTE), que

permite estimar el gasto térmico teórico (GTT) y el gasto térmico requerido (GTR) en función de la zona climática, las características constructivas de la vivienda y el tipo de instalaciones. Por otro, el módulo eléctrico, que da lugar al gasto eléctrico teórico (GELT) y que estima el consumo eléctrico esperado a partir del equipamiento del hogar, su ocupación y los patrones de uso de los distintos aparatos. Ambos módulos se integran en la herramienta DIAGNÓSTICO, que constituye el modelo de referencia de la Cátedra para el análisis del gasto energético en hogares españoles.

Hasta ahora, este modelo ha estado implementado en Engineering Equation Solver (EES), una herramienta que resultó adecuada en las etapas iniciales del desarrollo, especialmente para la validación numérica y el ajuste de los cálculos térmicos. Sin embargo, a medida que los estudios de la Cátedra han ido creciendo en escala y complejidad, sus limitaciones se han vuelto cada vez más evidentes: la dificultad para automatizar análisis sobre grandes volúmenes de datos, la poca flexibilidad para incorporar mejoras o nuevos módulos, y la escasa capacidad de integración con otros sistemas o herramientas digitales. En un contexto donde se manejan bases de datos cada vez más amplias y se demandan soluciones más dinámicas y reproducibles, estas restricciones representan un obstáculo práctico real para el trabajo de la Cátedra.

La necesidad de superar estas limitaciones es el punto de partida de este trabajo, pero su motivación va más allá de un cambio tecnológico. Migrar el modelo a un entorno más flexible y escalable tiene implicaciones concretas en distintos planos.

Desde el punto de vista social, mejorar la capacidad de diagnóstico energético de los hogares vulnerables es un paso necesario para que las ayudas existentes se ajusten mejor a las necesidades reales de cada familia. Un modelo que pueda aplicarse de forma sistemática a grandes bases de datos permite identificar con mayor precisión qué hogares se encuentran en situación de pobreza energética y en qué medida, algo que los indicadores estadísticos convencionales no siempre logran reflejar.

Desde el punto de vista económico, disponer de una estimación fiable del gasto energético teórico permite detectar qué hogares están por debajo de un umbral mínimo de consumo, con implicaciones tanto para el diseño de subsidios y ayudas públicas como para que las propias familias puedan entender mejor su situación y tomar decisiones más informadas.

Desde el punto de vista tecnológico, la migración del modelo a Python supone un avance práctico: permite automatizar cálculos, procesar grandes volúmenes de datos y abrir la puerta a su integración con otras herramientas digitales. Un entorno abierto y programable facilita además el mantenimiento del modelo, su reproducibilidad y su posible adopción por parte de otros equipos o instituciones con objetivos similares.

El presente Trabajo de Fin de Grado se enmarca dentro de esta línea de trabajo de la Cátedra de Energía y Pobreza, orientada al desarrollo de herramientas que permitan mejorar el diagnóstico energético de los hogares españoles. Disponer de modelos fiables, accesibles y escalables para estimar las necesidades energéticas reales de los hogares vulnerables es un paso necesario tanto para orientar la intervención

directa como para apoyar el diseño de políticas públicas más efectivas.

1.2. OBJETIVOS DEL PROYECTO

La situación descrita plantea una necesidad clara: disponer de una herramienta más flexible, escalable y accesible que permita seguir avanzando en el diagnóstico de la pobreza energética. De esta necesidad nace el objetivo principal de este Trabajo de Fin de Grado: migrar el modelo de cálculo del gasto energético requerido, actualmente implementado en EES, a un entorno Python que permita ampliar sus capacidades y facilitar su integración en herramientas digitales modernas. Para ello se plantean los siguientes objetivos:

- Replicar en Python el comportamiento del modelo original, trasladando sus ecuaciones y procedimientos y garantizando que los resultados obtenidos sean coherentes con los de EES.
- Diseñar una interfaz de usuario clara e intuitiva, que guíe al usuario a través del proceso de introducción de datos de forma lógica y pueda ser utilizada tanto por técnicos como por cualquier persona sin conocimientos especializados.
- Preparar el modelo para su aplicación en análisis de gran escala, de forma que pueda utilizarse para estudiar escenarios, evaluar medidas de intervención y apoyar el diseño de políticas públicas contra la pobreza energética.
- Desarrollar una aplicación web que implemente el modelo, permitiendo que cualquier usuario pueda estimar el gasto energético de su hogar de forma accesible y sin necesidad de conocimientos técnicos.

Capítulo 2. Estado de la Técnica

Este capítulo y el siguiente se basan principalmente en cuatro informes de la Cátedra de Energía y Pobreza[1]

Antes de describir los modelos, conviene aclarar tres conceptos que aparecen a lo largo de todo el trabajo. Según lo establece el CTE, la **demanda energética** es la energía que los sistemas técnicos de un edificio tendrían que ser capaces de proporcionar para mantener unas condiciones interiores adecuadas [9]. Se puede desglosar según el uso al que se destine: calefacción, refrigeración, agua caliente sanitaria (ACS) y electricidad no térmica. Este último término agrupa todo el consumo eléctrico que no está vinculado a la climatización ni al ACS: electrodomésticos, iluminación y dispositivos electrónicos. El **consumo energético** es la energía que hay que suministrar realmente a los sistemas para cubrir esa demanda, teniendo en cuenta la eficiencia de los equipos instalados. Por último, el **gasto energético** es el coste económico (en euros) de ese consumo, una vez aplicadas las tarifas vigentes.

Los tres conceptos se encadenan de la siguiente forma : la demanda determina la energía que el edificio necesita, el consumo traduce esa necesidad en energía suministrada al sistema, y el gasto expresa el coste económico de ese consumo.

2.1. MODELOS DE GASTO ENERGÉTICO EN HOGARES

En los últimos años, el diagnóstico de la pobreza energética y la estimación del gasto necesario en los hogares han impulsado el desarrollo de distintas herramientas orientadas a representar el comportamiento energético de las viviendas. Estas soluciones se apoyan en la normativa vigente, concretamente el Código Técnico de la Edificación (CTE) y la Calificación de la Eficiencia Energética de los Edificios (CEEE), y permiten calcular las necesidades asociadas a la calefacción, el agua caliente sanitaria (ACS), la refrigeración y los usos eléctricos no térmicos.

En este contexto, la Cátedra de Energía y Pobreza de la Universidad Pontificia Comillas ha desarrollado, en colaboración con ECODES, una metodología propia que ha ido evolucionando progresivamente para cubrir con mayor precisión las necesidades energéticas de los hogares españoles.

[1]Concretamente: *Caracterización del comportamiento energético en una muestra de hogares españoles* [2]; *Desarrollo de un modelo de cálculo de gasto eléctrico teórico en los hogares españoles* [3]; *Estudio del gasto térmico teórico en refrigeración para hogares españoles* [4]; e *Investigación aplicada para el desarrollo de una herramienta web de cálculo del gasto eléctrico teórico para hogares españoles* [1].

2.1.1. Gasto térmico teórico y gasto requerido

El punto de partida de los modelos desarrollados por la Cátedra de Energía y Pobreza es el concepto de **gasto térmico teórico (GTT)**, entendido como el coste que debería asumir un hogar para mantener unas condiciones de confort adecuadas dados su ubicación, superficie, tipología constructiva, nivel de aislamiento, número de ocupantes y tipo de instalaciones. A partir de esta definición, la Estrategia Nacional contra la Pobreza Energética 2019-2024 introdujo el concepto de **gasto térmico requerido (GR)**, una variante que fija un hogar de referencia estándar (100 m^2, 3 ocupantes, aislamiento deficiente, caldera individual de gas natural), tomando únicamente la ubicación real del hogar como variable. Este indicador sirve como umbral de referencia para calcular los indicadores nacionales de pobreza energética, aunque no representa el coste real de un hogar que no cumpla esas características.

El GTT se calcula a partir de la zona climática de la localidad, que se determina combinando la capital de provincia y la altitud. Con esa información, se toman las demandas de referencia por zona que recoge la CEEE y se corrigen según el aislamiento del edificio, que depende de cuándo se construyó y de su calificación energética, y según si la vivienda es unifamiliar o está en un bloque. En la práctica, esto significa que dos hogares en la misma ciudad pueden tener gastos térmicos teóricos muy distintos si uno es un piso en un edificio de los años 60 y el otro una casa construida después de 2007.

2.1.2. Gasto eléctrico teórico

En paralelo al módulo térmico, la Cátedra desarrolló en 2020 el **gasto eléctrico teórico (GELT)** [3], un modelo para estimar el consumo eléctrico esperado a partir del equipamiento del hogar y los patrones de uso de los distintos aparatos. El modelo distingue entre electrodomésticos cuyo uso varía a lo largo del día (cocina, lavadora, secadora, lavavajillas, televisión y ordenador) y aquellos con un perfil más estable (frigorífico, congelador, microondas, móvil, tableta e iluminación), aplicando métodos distintos en cada caso. Para los primeros, los patrones de uso se extraen de la Encuesta de Empleo de Tiempo (EET) del INE , realizada a 9.541 hogares en 2010, que permite construir curvas de potencia media en intervalos de diez minutos. Para los segundos, se usan estimaciones normativas del modelo simplificado previo. Una versión posterior, el Modelo Avanzado 2.0 [1], incorporó la ocupación de los miembros del hogar como variable adicional, dado que la presencia o ausencia de personas en la vivienda durante el día tiene un efecto apreciable sobre el consumo. Las categorías consideradas incluyen ocupado/a, parado/a, estudiante, jubilado/a, pensionista por incapacidad permanente, pensionista por viudedad u orfandad y personas que realizan tareas del hogar.

2.2. HERRAMIENTAS DE MODELADO ENERGÉTICO

Estos desarrollos se integraron en la herramienta DIAGNÓSTICO, que constituyó el primer entorno unificado para el análisis conjunto del gasto térmico y eléctrico, implementado en EES (*Engineering Equation Solver*). EES es un software comercial orientado a la resolución de sistemas de ecuaciones no lineales y algebraicos. Esas características lo hacían cómodo para traducir directamente las ecuaciones de la normativa sin necesidad de cálculos manuales, lo que facilitó la fase inicial de desarrollo y validación del modelo.

Sin embargo, el debate sobre la pobreza energética había estado centrado tradicionalmente en la calefacción y el ACS, dejando fuera la refrigeración. Para dar respuesta a esta laguna, la Cátedra amplió el modelo en 2022 con un módulo específico para estimar el gasto térmico en refrigeración [4], incorporando demandas de referencia por zona climática de verano, factores de corrección relacionados con el aislamiento del edificio y rendimientos estacionales de los equipos. Esta extensión reflejó además un cambio en el contexto: el aumento de las olas de calor y la creciente penetración del aire acondicionado en los hogares españoles hacían ya difícil ignorar este uso energético en el diagnóstico de la vulnerabilidad. Según datos del INE, en 2008 solo el 35,5 % de los hogares españoles disponía de aire acondicionado, pero esa cifra ha ido creciendo de forma sostenida.

A partir de esta base, se desarrolló la plataforma web ENERSOC, que migra el núcleo de cálculo a un entorno en línea accesible a profesionales del ámbito social. Sin embargo su implementación fue parcial y no cubre la totalidad de los módulos de DIAGNÓSTICO.

Cuando el modelo se aplicó a la base de datos del programa «Ni un hogar sin energía», el consumo real de los hogares vulnerables resultó entre un 18 % y un 20 % inferior al teórico, lo que indica que una parte significativa de la muestra no cubría sus necesidades energéticas mínimas. Este resultado también puso de manifiesto una limitación práctica: EES no dispone de API, ni se integra con bases de datos o entornos web, y sus archivos no son ejecutables fuera del software propio. A esto se añade que es una herramienta de licencia propietaria. En la práctica, esto significa que aplicar el modelo de forma sistemática a grandes volúmenes de datos, o integrarlo en una plataforma web, no es viable con la implementación actual. Estas limitaciones son las que motivan la migración del modelo a Python, que se describe en los capítulos siguientes.

2.3. ESTRUCTURA DE LA HERRAMIENTA DIAGNÓSTICO

La herramienta DIAGNÓSTICO está formada por tres módulos de cálculo: el módulo térmico de calefacción y ACS, el módulo de refrigeración y el módulo eléctrico.

Los tres parten de las mismas entradas básicas: localización del hogar, superficie, tipo de vivienda, época de construcción, nivel de aislamiento, tipo de instalaciones y composición del hogar. A partir de estas entradas, cada módulo calcula la demanda energética correspondiente, la convierte en consumo aplicando niveles de eficiencia y obtiene el gasto final a partir de las tarifas correspondientes.

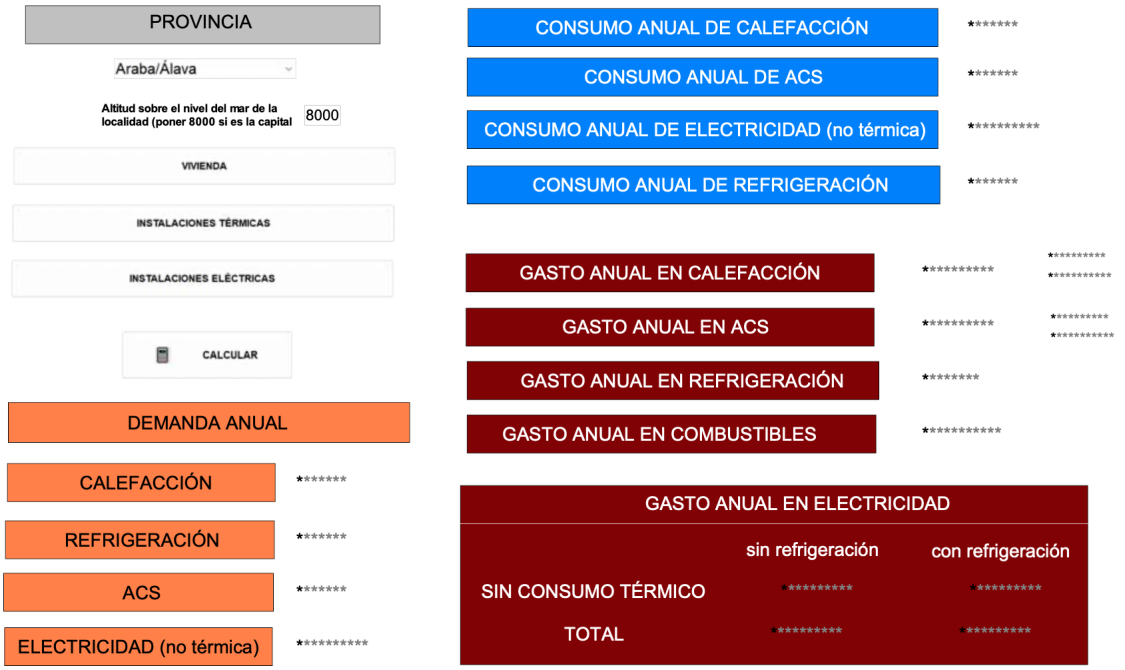
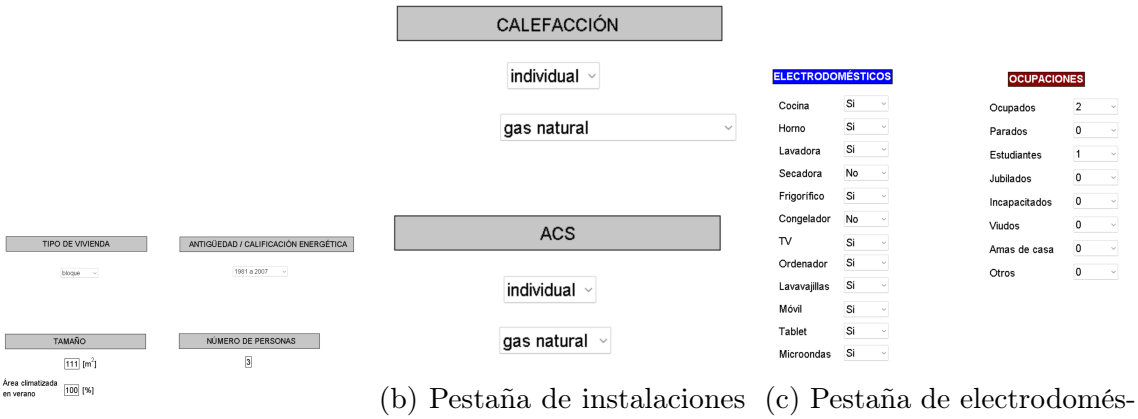


Figura 2.1: Pantalla principal de DIAGNÓSTICO en EES, con los campos de entrada a la izquierda y las salidas a la derecha.



(a) Pestaña de vivienda.

(b) Pestaña de instalaciones térmicas.

(c) Pestaña de electrodomésticos y perfil del ocupante.

Figura 2.2: Pestañas de entrada de datos de la herramienta DIAGNÓSTICO en EES.

2.3.1. Módulo térmico: calefacción y ACS

Calefacción

Demanda

Este módulo se construye sobre la siguiente idea general: calcular cuánto le costaría a un hogar mantener una temperatura de confort en invierno (21°C según el CTE) y tener agua caliente, dados su ubicación, su edificio y sus instalaciones.

Para ello combina dos tipos de información. Por un lado, datos normativos sobre cuánta energía necesita típicamente un edificio según dónde está y cómo está construido, y por otro, las condiciones reales de la vivienda analizada.

El punto de partida es la zona climática de invierno [2] (ZCI), que el modelo asigna a partir de la capital de provincia y la altitud de la localidad. Esto determina la demanda de referencia D_R. A partir de ahí, el modelo ajusta esa referencia en función de dos factores.

El primero es el factor de corrección f_c, que ajusta la demanda según el nivel de aislamiento térmico del edificio.

$$f_c = \begin{cases} \dfrac{1 + (C_1 - 0{,}6) \cdot 2 \cdot (R - 1)}{R} & \text{edificios nuevos} \\ \dfrac{1 + (C_2 - 0{,}5) \cdot 2 \cdot (R' - 1)}{R'} & \text{edificios existentes} \end{cases} \tag{2.1}$$

donde R y R' son las dispersiones de las demandas de referencia para edificios nuevos y existentes, definidas en la CEEE según la zona climática y el tipo de vivienda.

Los índices C_1 y C_2 caracterizan el nivel de aislamiento del edificio. El modelo los asigna según la época de construcción, que determina qué normativa térmica estaba en vigor. Una vivienda construida antes de 1980, sin exigencias normativas de aislamiento, presenta una demanda energética significativamente mayor que una construida después de 2006 bajo el CTE.

Los edificios anteriores a 1980 muestran las peores prestaciones térmicas (Aislamiento deficiente) (calificación F-G); los construidos entre 1980 y 2006, regulados por la NBE, presentan un comportamiento intermedio (Aislamiento normal) (calificación E-F); mientras que los posteriores a 2006 son los más eficientes (Aislamiento CTE)(calificación D-E). No obstante, el modelo les asigna un valor más bajo, al considerar que gran parte de las viviendas analizadas pertenecen a hogares vulnerables que incluso en edificios más recientes, suelen corresponder a viviendas con niveles de eficiencia más bajos.

[2] La letra de la zona climática indica la severidad del invierno (de la A:más suave, a la E:más fría) y el número la severidad del verano (de 1 a 4).

El segundo factor es la tipología de la vivienda. Los bloques tienen menos superficie en contacto con el exterior que las viviendas unifamiliares, por lo que sus demandas de referencia son menores.

La demanda final D se obtiene multiplicando la demanda de referencia [kWh/m²] por la superficie de la vivienda A y el factor de corrección f_c:

$$D = D_R \cdot A \cdot f_c \tag{2.2}$$

Consumo

En la misma línea, el consumo se obtiene dividiendo la demanda entre el rendimiento medio estacional del sistema de calefacción (SPF):

$$C = \frac{D}{\text{SPF}} \tag{2.3}$$

El valor del SPF varía según el sistema: una caldera de gas natural tiene mejor rendimiento que una de gasóleo o carbón, y una bomba de calor mejor aún. Además, el modelo distingue entre instalaciones individuales, centrales y por aparatos, asignando un SPF diferente a cada una. En general, las instalaciones centrales presentan un rendimiento menor que las individuales, debido a las pérdidas asociadas a la distribución del calor.

Gasto

Para convertir el consumo en gasto se aplican las tarifas comerciales de cada vector energético (gas natural, GLP o electricidad.), a las que se añaden los impuestos y los términos fijos correspondientes.

$$G = \left(C \cdot \text{tarifa}_{\text{variable}} + \text{tarifa}_{\text{fijo}}\right) \cdot (1 + \text{IVA}) \tag{2.4}$$

donde C es el consumo en kWh, $\text{tarifa}_{\text{variable}}$ es el precio por kWh del vector energético correspondiente, $\text{tarifa}_{\text{fijo}}$ recoge los términos fijos de la factura (alquiler de contador e impuestos fijos) y el factor $(1 + \text{IVA})$ aplica los impuestos correspondientes según la ubicación.

En el caso de los combustibles no eléctricos (gas natural, GLP, gasóleo, biomasa y carbón), el gasto se calcula de forma similar pero con una estructura de costes diferente. Por un lado, no hay término fijo de potencia contratada; en su lugar, el gas natural en instalación individual incluye una cuota mensual de alquiler del contador. Por otro lado, el término variable es el precio del combustible en €/kWh multiplicado

por el consumo. A esto se añade el impuesto sobre los hidrocarburos, que solo aplica al gas natural, y el IVA, cuyo tipo varía según el combustible y la ubicación del hogar. En el caso del GLP y el gasóleo, las tarifas tienen variantes específicas para Canarias, Ceuta y Melilla.

$$G = \left(\text{tarifa}_{\text{fijo}} \cdot 12 + \text{tarifa}_{\text{variable}} \cdot C + \text{alquiler} \cdot 12 + \text{imp}_{\text{hidro}} \cdot C\right) \cdot (1 + \text{IVA}) \quad (2.5)$$

donde $\text{tarifa}_{\text{fijo}}$ es el término fijo mensual de la tarifa del combustible, alquiler la cuota mensual de alquiler del contador (solo para gas natural en instalación individual) y $\text{imp}_{\text{hidro}}$ el impuesto sobre los hidrocarburos en €/kWh, aplicable únicamente al gas natural.

Los valores concretos de cada término (precio por kWh, término fijo, impuesto y alquiler de contador) se recogen en la tabla *tarifas* e *impuestos*.

Agua caliente sanitaria (ACS)

Demanda

La demanda de ACS se calcula de forma independiente a la de calefacción; no depende del clima sino del número de personas que viven en la vivienda y de la temperatura del agua de red en la localidad, que varía considerablemente de unas zonas a otras.

Los valores de temperatura mensual del agua de red para las capitales de provincia, $T_{\text{capital},i}$, están establecidos según el CTE. Cuando la localidad no coincide con la capital de provincia, se corrige por altitud:

$$T_{\text{localidad},i} = T_{\text{capital},i} - B_i \cdot (H_{\text{localidad}} - H_{\text{capital}}) \quad (2.6)$$

donde B_i vale 0,0066 en los meses de octubre a marzo y 0,0033 de abril a septiembre.

Con esa temperatura corregida y el número de ocupantes, el CTE establece un consumo normativo de 28 l/persona·día a 60 °C, (para viviendas en bloque se aplica un factor corrector de 22/30), lo que da una demanda anual de:

$$D_{\text{ACS}}\,[\text{kWh}] = N_{pax} \cdot Q \cdot \frac{365}{12} \cdot 4{,}176 \cdot \frac{12 \cdot 60 - \sum_{i=1}^{12} T_{\text{red},i}}{3\,600} \quad (2.7)$$

donde N_{pax} es el número de ocupantes, $Q = 28\,\text{l/persona·día}$ (28), $4{,}176\,\text{kJ/(kg·K)}$ es el calor específico del agua y $T_{\text{red},i}$ es la temperatura del agua de red en el mes i [°C].

En la práctica, esto significa que un hogar en Burgos necesita calentar el agua de red bastante más que uno en Almería, donde el agua de red llega a temperaturas más altas durante casi todo el año.

Consumo y gasto

Una vez obtenida la demanda, el consumo y el gasto se calculan igual que la calefacción: dividiendo por el SPF del sistema de ACS y aplicando la tarifa correspondiente.

2.3.2. Módulo de refrigeración

Demanda

El módulo de refrigeración mantiene la misma estructura de cálculo que el de calefacción, pero aplicada a las condiciones de verano.

El primer paso consiste en asignar a cada localidad su zona climática de verano (ZCV), con valores de 1 a 4, siendo 4 la máxima intensidad de las condiciones climáticas en verano. Las zonas con ZCV igual a 1 se consideran que no tienen demanda de refrigeración. Una vez conocida la zona climática, la demanda de referencia D_R [kWh/m^2] se obtiene de las tablas de la CEEE con valores diferenciados para viviendas unifamiliares y en bloque. Esa referencia se corrige mediante el factor f_c:

$$f_c = \frac{1 + (C_1 - 0{,}6) \cdot 2 \cdot (R - 1)}{R} \tag{2.8}$$

,

usando siempre los valores para edificios de nueva construcción (C_1 y R), que es el procedimiento que establece la CEEE para refrigeración independientemente de la antigüedad del edificio. Se incorpora además un factor f_s que representa el porcentaje de superficie climatizada en verano, dado que lo habitual es refrigerar solo algunas estancias.

Pues, la demanda final D se calcula como:

$$D = D_R \cdot f_s \cdot A \cdot f_c \tag{2.9}$$

donde A es la superficie de la vivienda en m^2.

Consumo

El consumo eléctrico se obtiene dividiendo la demanda entre el SEER del equipo. El modelo asume equipos de aire acondicionado eléctricos en configuración partida, que representan el 92 % de los sistemas de climatización residenciales en España [4].

$$C = \frac{D}{\text{SEER}} \tag{2.10}$$

Los valores de SEER dependen de la zona climática de verano y distinguen entre edificios existentes y de nueva construcción

Gasto

Para convertir el consumo en gasto se aplica la tarifa PVPC 2.0 TD, que tiene dos componentes.

El término fijo depende de la potencia contratada y se paga con independencia del consumo:

$$\text{tarifa}_{\text{fijo}} = \frac{\text{fijo}_{\text{punta}} \cdot P_{\text{punta}} + \text{fijo}_{\text{valle}} \cdot P_{\text{valle}}}{P_{\text{punta}}} \tag{2.11}$$

El término variable depende de los kWh consumidos. Dado que el consumo de refrigeración se concentra en verano y en las horas de más calor, que coinciden con las franjas punta y llano de la tarifa, se pondera mediante unos pesos que reflejan esa distribución horaria según la zona climática:

$$\text{tarifa}_{\text{variable}} = \text{tarifa}_{\text{punta}} \cdot peso_{\text{punta}} + \text{tarifa}_{\text{llano}} \cdot peso_{\text{llano}} + \text{tarifa}_{\text{valle}} \cdot peso_{\text{valle}} \tag{2.12}$$

El gasto de refrigeración se calcula aplicando el término variable de la tarifa eléctrica al consumo estimado. El término fijo asociado a la potencia contratada no se incluye en este módulo ya que el posible aumento de potencia por disponer de aire acondicionado (de 5,0 a 6,9 kW en punta) se tiene en cuenta dentro del módulo eléctrico, donde se compara el coste con y sin refrigeración.

$$G_{\text{REF}} = C \cdot \text{tarifa}_{\text{variable}} \cdot (1 + \text{imp}_{\text{elec}}) \cdot (1 + \text{IVA}) \tag{2.13}$$

2.3.3. Módulo eléctrico

El módulo eléctrico estima el consumo asociado a los usos no térmicos del hogar: electrodomésticos, iluminación y dispositivos electrónicos. Es el más complejo de los

tres porque no parte de normativa constructiva sino de datos sobre comportamiento real de los hogares.

Consumo

El módulo eléctrico estima el consumo eléctrico del hogar a partir del equipamiento, el número de ocupantes, la superficie de la vivienda y la ocupación de sus miembros.

El modelo distingue dos grupos de aparatos según cómo se usa cada uno.

El primero lo forman los electrodomésticos cuyo uso varía a lo largo del día: cocina, lavadora, secadora, lavavajillas, televisión y ordenador. Para estos, el modelo utiliza perfiles de uso obtenidos de la Encuesta de Empleo del Tiempo (EET), que permiten distribuir el consumo en intervalos de diez minutos a lo largo del día.

El consumo de cada aparato j se calcula a partir de su potencia media en cada intervalo de diez minutos a lo largo del día y del número de días del año (aproximación de 12 meses de 30 días):

$$C_{jlrs} = \left(\sum_{k=1}^{144} P_{jkl} \right) \cdot \frac{10}{60} \cdot 360 \quad [\text{kWh}] \tag{2.14}$$

donde $P_{j,k,l,rs}$ es la potencia media del aparato j en el intervalo k, para un hogar de tamaño l y ocupación rs.

El segundo grupo lo forman los aparatos con un perfil de consumo más estable: frigorífico, congelador, microondas, iluminación y dispositivos móviles. Para estos se utilizan estimaciones medias de consumo ajustadas según el tamaño de la vivienda y el número de ocupantes, ya que su consumo no varía de forma apreciable según los hábitos del hogar.

A partir de los consumos individuales de cada aparato se definen unos factores de corrección por ocupación f_{lrs} que representan cuánto sube o baja el consumo de ese tipo de hogar respecto al consumo medio de un hogar del mismo tamaño sin distinción de ocupación.

$$f_{lrs} = C_{\text{total},6,lrs} - C_{\text{total},6,l} \tag{2.15}$$

Estos ajustes reflejan que los patrones de consumo cambian según la actividad principal de los miembros del hogar. Un hogar con personas jubiladas o desempleadas suele pasar más tiempo en casa que uno formado principalmente por personas ocupadas.

El consumo total resulta de sumar el consumo de cada aparato disponible y los factores de ajuste de cada miembro:

$$C = \sum_{j=1}^{12} C_{jl} \cdot \beta_j + \sum_{i=1}^{l} f_{lrsi} \cdot \alpha_{lrsi} \tag{2.16}$$

donde β_j vale 1 si el hogar dispone del aparato j y 0 si no, y α_i vale 1 si el miembro i corresponde a la ocupación rs y 0 si no.

Si se desconoce si un hogar dispone de un aparato concreto, se asume un consumo medio ponderado por la tasa de penetración del aparato en hogares de tamaño similar, en lugar del consumo individual.

Gasto

Para convertir el consumo eléctrico en gasto (GELT), el modelo aplica la tarifa PVPC 2.0 TD, que distingue tres periodos horarios: punta (de 10:00 a 14:00 y de 18:00 a 22:00), llano (de 8:00 a 10:00 y de 14:00 a 18:00) y valle (de medianoche a 8:00 en días laborables y durante todo el día en fines de semana y festivos). Para el término fijo de potencia y el término variable de consumo se utilizan valores medios asociados a cada franja horaria.

Por un lado, el término fijo depende de la potencia contratada y sigue la misma expresión que la ecuación 2.11

Por otro lado, el término variable se pondera según la distribución del consumo doméstico a lo largo del día, con unos pesos de 29 % en punta, 26 % en llano y 45 % en valle, siguiendo la misma expresión que la ecuación 2.12.

El gasto eléctrico anual se obtiene aplicando la tarifa al consumo e incluyendo el impuesto sobre la electricidad y el IVA:

$$G_{\text{elec,noTérmico}} = (\text{tarifa}_{\text{fijo}} \cdot 12 + \text{tarifa}_{\text{variable}} \cdot C) \cdot (1 + \text{imp}_{\text{elec}}) \cdot (1 + \text{IVA}) + 0{,}81 \cdot 12 \cdot (1 + \text{IVA}) \tag{2.17}$$

donde T_{fijo} es el término fijo mensual, C el consumo eléctrico anual no térmico, e imp_{elec} el impuesto sobre la electricidad, fijado en el 5,11 % . El factor $0{,}81 \cdot 12$ corresponde al alquiler anual del equipo; un cargo fijo que aparece en todas las facturas independientemente del consumo o la potencia contratada.

2.3.4. Salidas del modelo

Una vez calculados los gastos de cada módulo, el modelo los agrupa según el tipo de suministro en dos conceptos: el **gasto eléctrico**, que suma todos los usos que van a la factura de la luz, y el **gasto en combustibles**, que recoge lo que se paga en gas natural, GLP, gasóleo, biomasa y carbón. Si tanto la calefacción como el ACS son eléctricas, este segundo término es cero.

En conjunto, el modelo devuelve las siguientes salidas para cada hogar analizado:

- **Demanda** [kWh]: energía que el edificio necesita para calefacción, refrigeración, ACS y usos eléctricos no térmicos.
- **Consumo** [kWh]: energía que hay que suministrar realmente a los sistemas.
- **Gasto** [€]: coste económico anual desglosado en calefacción, ACS, refrigeración, electricidad no térmica y combustibles.

La demanda y el consumo se desglosan en cuatro usos: calefacción, refrigeración, ACS y electricidad no térmica. El gasto, en cambio, se agrupa por vector energético: todo lo que va a la factura de la luz forma el gasto eléctrico total, y lo que se paga aparte en gas natural, GLP, gasóleo, biomasa o carbón constituye el gasto en combustibles. Esta separación permite identificar qué parte del gasto depende de la tarifa eléctrica y qué parte de los precios del gas o el gasóleo.

El gasto eléctrico conjunto se calcula de forma distinta al gasto individual de cada uso térmico. Mientras que los gastos de calefacción, ACS y refrigeración se obtienen de forma independiente —aplicando a cada uno su propia tarifa y términos fijos por separado—, el gasto eléctrico conjunto refleja lo que aparecería en una única factura con todos los consumos agregados. La diferencia práctica es que el término fijo de potencia se paga una sola vez, y la potencia contratada puede variar según si el hogar dispone o no de refrigeración.

Por eso el modelo calcula este gasto en dos variantes. En la variante sin refrigeración, la potencia contratada se determina a partir del número de ocupantes y del sistema de calefacción. En la variante con refrigeración, si el porcentaje de superficie climatizada en verano supera el 30 %, la potencia contratada puede aumentar de 5,0 a 6,9 kW, salvo que el hogar ya requiriese una potencia mayor por otros usos. En localidades con zona climática de verano igual a 1, ambas variantes coinciden.

El gasto eléctrico no térmico se calcula según la ecuación 2.17, usando en cada caso la potencia contratada correspondiente. A partir de ahí, el gasto eléctrico total suma los usos térmicos que funcionan con electricidad:

$$G_{\text{elec,TOTAL,sinREF}} = G_{\text{elec,noTérmico,sinREF}} + G_{\text{ACS}} + G_{\text{CAL}} \tag{2.18}$$

$$G_{\text{elec,TOTAL,conREF}} = G_{\text{elec,noTérmico,conREF}} + G_{\text{ACS}} + G_{\text{CAL}} + G_{\text{REF}} \tag{2.19}$$

donde G_{ACS} y G_{CAL} solo se incluyen si el sistema correspondiente es eléctrico; si funciona con gas natural, GLP o gasóleo, ese gasto pasa al término de combustibles.

El gasto en combustibles recoge únicamente los usos térmicos no eléctricos:

$$G_{\text{comb}} = G_{\text{ACS,comb}} + G_{\text{CAL,comb}} \tag{2.20}$$

donde $G_{\text{ACS,comb}}$ y $G_{\text{CAL,comb}}$ son el gasto de ACS y calefacción en combustible y valen cero si el sistema correspondiente es eléctrico. En caso contrario, incluyen los términos fijos, el coste variable por kWh consumido, el impuesto sobre el gas y el IVA correspondiente.

2.3.5. Datos de entrada y tablas de referencia

La herramienta DIAGNOSTICO requiere dos tipos de entradas. Por un lado, los datos del hogar que introduce el usuario:

- Provincia y altitud de la localidad
- Antigüedad o calificación energética del edificio
- Tipo de vivienda y superficie en m^2
- Número de ocupantes y ocupación laboral de cada miembro
- Porcentaje de superficie climatizada en verano
- Tipo de instalación de calefacción y de ACS
- Fuente de energía de calefacción y ACS
- Equipamiento eléctrico disponible en el hogar

Por otro lado, consta de un conjunto de tablas que sirven como bases de datos para realizar todos los cálculos.

Zonas climáticas

La tabla *zonas* asigna a cada combinación de provincia y altitud su zona climática, expresada como un código de letra y número. La letra indica la severidad del invierno (de la A a la E) y el número la del verano (de 1 a 4).

Demandas de referencia

Las tablas *sci_referencia* y *scv_referencia* contienen las demandas de referencia de calefacción y refrigeración respectivamente [kWh/m^2], diferenciadas por zona climática y tipología de vivienda, según la CEEE. Son el punto de partida para calcular la demanda de cada vivienda antes de aplicar los factores de corrección.

Índices de calificación y dispersión

Los valores de C_1 y C_2 se obtienen de las tablas *C1_bloque*, *C1_unifamiliar*, *C1_bloque_verano* y *C1_unifamiliar_verano* clasificadas según la calificación energética y la zona climática de invierno y verano, respectivamente. Estos índices se usan para calcular el factor de corrección f_c de la ecuación 2.1. Asimismo, las dispersiones R y R' se obtienen de *Dispersion R* y *Dispersion R verano*.

Rendimientos estacionales

Las tablas *HSPF_nuevo* y *HSPF_existente* recogen el SPF de cada sistema de calefacción según el tipo de instalación y la antigüedad del edificio. Asimismo, *ACS_SPFnuevo* y *ACS_SPFexistente* contienen el SPF del sistema de ACS. La tabla *SEER* recoge los rendimientos de refrigeración por zona climática de verano, distinguiendo también entre edificios existentes y nuevos.

Temperatura del agua de red

La tabla *Temp red ACS* contiene la temperatura mensual del agua de red para cada capital de provincia y su altitud. Cuando la localidad no coincide con la capital, el modelo aplica la corrección por altitud de la ecuación 2.6.

Consumo eléctrico por electrodoméstico

Las tablas *CMESin* y *CMECon* recogen el consumo medio anual de cada aparato [kWh] según el número de miembros, sin y con el factor de penetración. La primera se usa cuando se sabe que el hogar dispone del aparato; la segunda, cuando se desconoce. La tabla *Penetracion* contiene las tasas de penetración de los aparatos que no están presentes en todos los hogares.

Factores de ajuste por ocupación

La tabla *Factores* contiene los factores f_{lrs} para cada combinación de ocupación laboral y tamaño de hogar. Valores positivos indican mayor consumo que el hogar medio de igual tamaño y negativos, menor.

Iluminación

El consumo de iluminación se obtiene de tablas específicas por zona geográfica (*Sur* y *Canarias*), organizadas por número de ocupantes y tramos de superficie. La zona de iluminación de cada provincia se determina a partir de la propia tabla *zonas*.

Tarifas e impuestos

La tabla *pvpc* contiene los términos fijo [€/kW·año] y variable [€/kWh] de la tarifa PVPC 2.0 TD por tramo horario. Los pesos de distribución del consumo entre franjas se leen de *pesos tarifa electrica* para usos domésticos y de *pesos tarifa electrica verano* para refrigeración. Las tarifas de combustibles (gas natural, GLP, gasóleo, carbón y biomasa) se recogen en *tarifas*, con variantes por territorio para GLP y gasóleo. Los impuestos se obtienen de *impuestos* y los costes de alquiler de equipos de *alquiler equipos*.

2.3.6. Flujo de cálculo

La siguiente figura muestra de forma esquemática el orden en que el modelo ejecuta los cálculos, desde los datos de entrada hasta la obtención de los resultados finales.

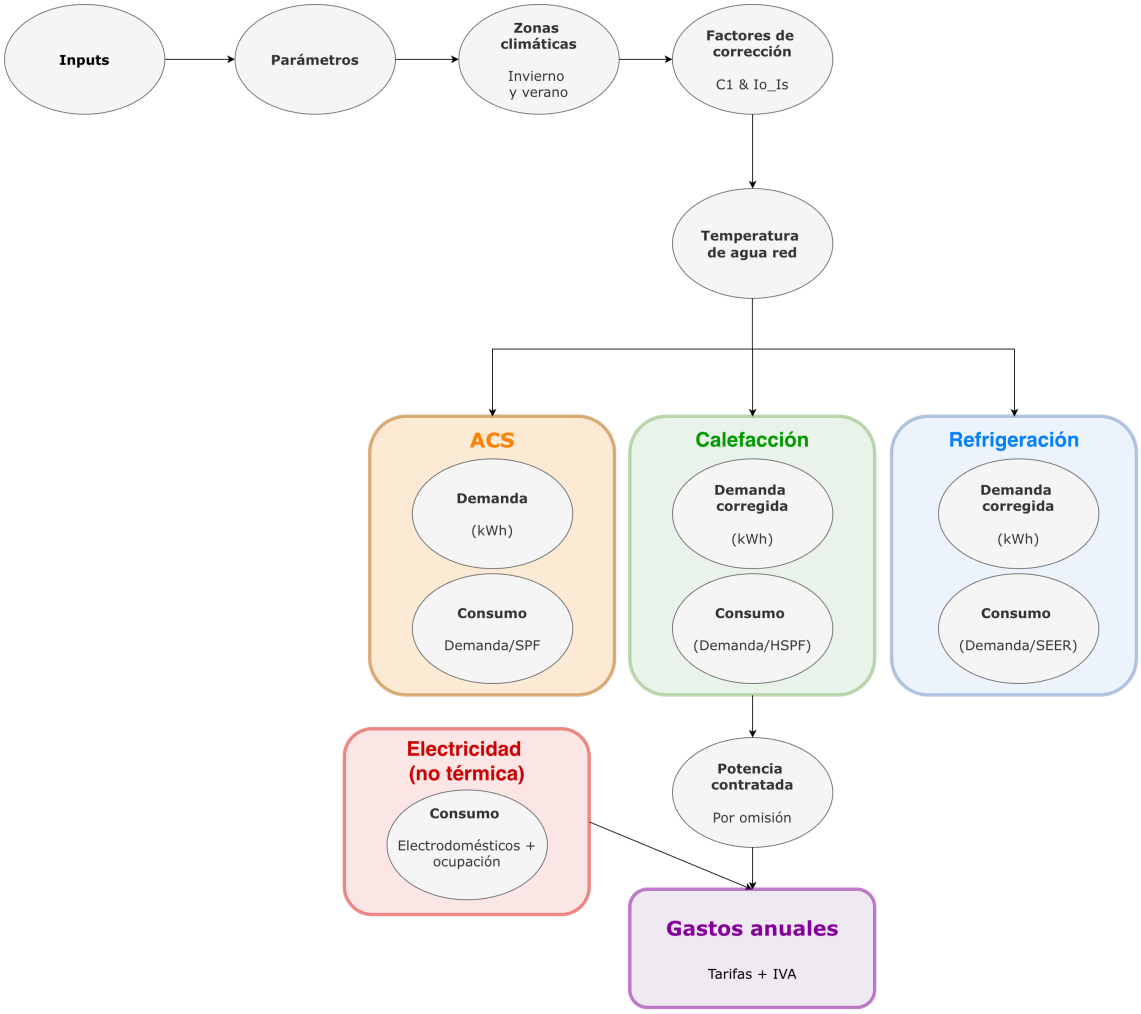


Figura 2.3: Secuencia de cálculo de la herramienta DIAGNÓSTICO.

Capítulo 3. Desarrollo Experimental

El objetivo de este capítulo es describir cómo se ha llevado a cabo la migración del modelo a Python, qué decisiones de diseño se han tomado y cómo se ha desarrollado la interfaz de usuario. La elección de Python responde a varias razones prácticas: es un lenguaje de uso libre, ampliamente extendido en el ámbito científico y de análisis de datos y con una gran variedad de librerías que facilitan tanto el cálculo numérico como el desarrollo de aplicaciones web.

3.1. PLANTEAMIENTO

El trabajo comenzó con un análisis previo del modelo implementado en EES. A raíz de este análisis el desarrollo se organizó en tres fases. En la primera se desarrolló la interfaz de usuario, definiendo los campos de entrada y la presentación de resultados. En la segunda se llevó a cabo la implementación de las funciones y el módulo de cálculo y en la tercera se coordinó el flujo entre la interfaz y el motor de cálculo.

El primer paso fue analizar en detalle el modelo original para entender su estructura y sus dependencias internas. El código EES se organiza como una serie de funciones que se llaman entre sí y acceden a tablas de datos internas mediante las funciones *Lookup* y *Lookup$Row*. Cada función recibe una serie de parámetros de entrada, consulta los valores necesarios en las tablas y devuelve un resultado que otras funciones utilizan.

Este análisis permitió identificar tres elementos que estructuran todo el modelo: las funciones de cálculo, las tablas de referencia que contienen los parámetros y datos utilizados, y la lógica del programa principal, que determina el orden en que se calculan las variables intermedias. Entender esa secuencia fue especialmente importante a la hora de implementarlo en Python, ya que ambos entornos difieren en la forma en que ejecutan los cálculos. EES resuelve todas las ecuaciones de forma simultánea, sin tener en cuenta el orden en que están definidas las variables. Python, en cambio, ejecuta el código de forma secuencial, por lo que cada variable debe estar definida antes de ser utilizada.

Una vez analizado el modelo, se decidió seguir un enfoque de traducción directa función a función. El código se desarrolló inicialmente en español, manteniendo los nombres de variables y funciones lo más próximos posible a los originales para facilitar la comparación con el modelo EES. Una vez validado el funcionamiento, se tradujo al inglés. Los textos visibles para el usuario y los identificadores utilizados para acceder a las tablas Excel se mantuvieron en español, ya que no podían modificarse

sin alterar el funcionamiento del modelo.

Con ese planteamiento definido, se identificaron cuatro áreas de responsabilidad: el acceso a los datos de las tablas de referencia, la lógica de cálculo energético, la coordinación del flujo entre la interfaz y el motor de cálculo, y la propia interfaz de usuario. Cada una de ellas se implementó como un módulo independiente, lo que permitió desarrollar y depurar cada parte por separado.

Los apartados siguientes describen las decisiones tomadas en cada una de ellas.

3.2. ESTRUCTURA

El código desarrollado durante este Trabajo Fin de Grado, junto con las tablas de referencia necesarias para ejecutar el modelo, se encuentra almacenado en un repositorio privado de GitHub. El repositorio contiene la implementación en Python del modelo DIAGNÓSTICO, la interfaz desarrollada con Streamlit y los ficheros de datos empleados para reproducir los cálculos. Dado que el repositorio incluye la implementación completa del modelo y las tablas necesarias para su funcionamiento, se ha mantenido en modo privado. La información relativa a la disponibilidad del código se recoge en el Anexo II, y el acceso puede solicitarse a través del correo universitario de la autora: `202202350@alu.comillas.edu`.

El código en Python se estructuró en cuatro ficheros principales, cada uno con una función concreta dentro del proyecto. Esta organización separa la interfaz de usuario, el flujo de datos , la lógica de cálculo energético y el acceso a las tablas de referencia, lo que facilita tanto el desarrollo como futuras modificaciones.

- `app.py` contiene toda la lógica de presentación. Define las cinco pestañas de entrada, el estilo visual mediante CSS personalizado y todos los elementos interactivos. Cuando el usuario pulsa el botón "Ejecutar Simulación", recoge los valores del estado de la sesión y los presenta en tres bloques: demanda energética anual, gasto anual en euros y consumo, desglosados por sistema.

- `help.py` gestiona el acceso a las tablas de referencia. Principalmente consta de dos funciones; `lookup_row()` y `lookup_value()`, que permiten consultar cualquier valor en cualquier tabla. Son el equivalente directo de *Lookup$Row* y *Lookup* en EES.

- `model.py` es el núcleo de cálculo. Contiene todas las funciones que implementan la lógica energética del modelo: determinación de zonas climáticas, cálculo de demandas, eficiencias de los sistemas térmicos, tarifas e impuestos. Replica el comportamiento del modelo EES original función a función. Todas las funciones siguen el mismo patrón: reciben sus parámetros, consultan los valores necesarios en las tablas mediante `help.py` y devuelven un único resultado.

- `main.py` coordina el flujo entre la interfaz y el motor de cálculo. Su función principal, `run_demo()`, recibe el diccionario de inputs, llama a las funciones de `model.py` y devuelve los resultados a `app.py`.

3.3. METODOLOGÍA

3.3.1. Interfaz de usuario

La primera fase del desarrollo fue la construcción de la interfaz de usuario. Se comenzó por la interfaz de usuario por ser la parte más clara desde el principio: las entradas del modelo estaban perfectamente definidas en EES, por lo que trasladarlas no requería tomar decisiones sobre la lógica de cálculo. Disponer de una interfaz funcional desde las primeras etapas del desarrollo permitió además tener una referencia visual del resultado final antes de abordar la implementación del motor de cálculo.

Para ello se utilizó Streamlit, una librería de Python orientada al desarrollo de aplicaciones de datos que permite construir interfaces interactivas sin necesidad de separar el proyecto en una capa de frontend y otra de backend. Frente a otras alternativas, Streamlit requiere menos configuración y permite centrarse en la lógica de la aplicación, lo que lo hacía más adecuado para los objetivos del proyecto.

La interfaz se diseñó desde el principio con una estructura de cinco pestañas, con el objetivo de agrupar los campos de entrada por temática y facilitar la navegación al usuario. Dado el volumen de parámetros que requiere el modelo, presentarlos todos en una única pantalla habría resultado poco manejable.

Las pestañas son las siguientes:

- **Datos Generales**: recoge la provincia y la altitud de la localidad.



Figura 3.1: Pestaña de Datos Generales.

- **Vivienda**: incluye el tipo de vivienda (bloque o unifamiliar), la antigüedad o calificación energética, la superficie en m^2 y el número de habitantes. También se introduce aquí el porcentaje de superficie climatizada en verano, que el modelo utiliza para escalar la demanda de refrigeración.

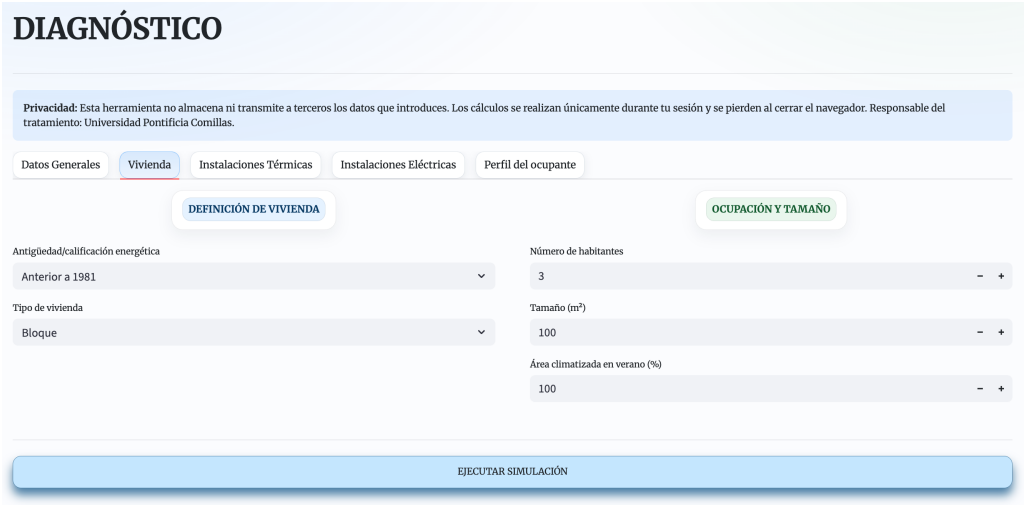


Figura 3.2: Pestaña de Vivienda.

- **Instalaciones Térmicas**: permite seleccionar el sistema de calefacción y el sistema de ACS de forma independiente, junto con el tipo de instalación de cada uno (individual, central o por aparatos). Esta separación es necesaria porque el modelo trata ambos sistemas de forma independiente y pueden tener combustibles y configuraciones distintas.



Figura 3.3: Pestaña de Instalaciones Térmicas.

- **Instalaciones Eléctricas**: recoge información sobre los electrodomésticos presentes en el hogar. Para cada uno de ellos el usuario puede indicar si está presente (*Sí*), si no lo está (*No*), o si no lo sabe (*NS/NC*). Esta distinción tiene

efecto directo en el cálculo: cuando se indica *Sí*, el modelo asigna el consumo completo del aparato; cuando se indica *No*, el consumo es cero; y cuando se indica *NS/NC*, el modelo pondera su consumo según la probabilidad de que un hogar de esas características lo tenga.

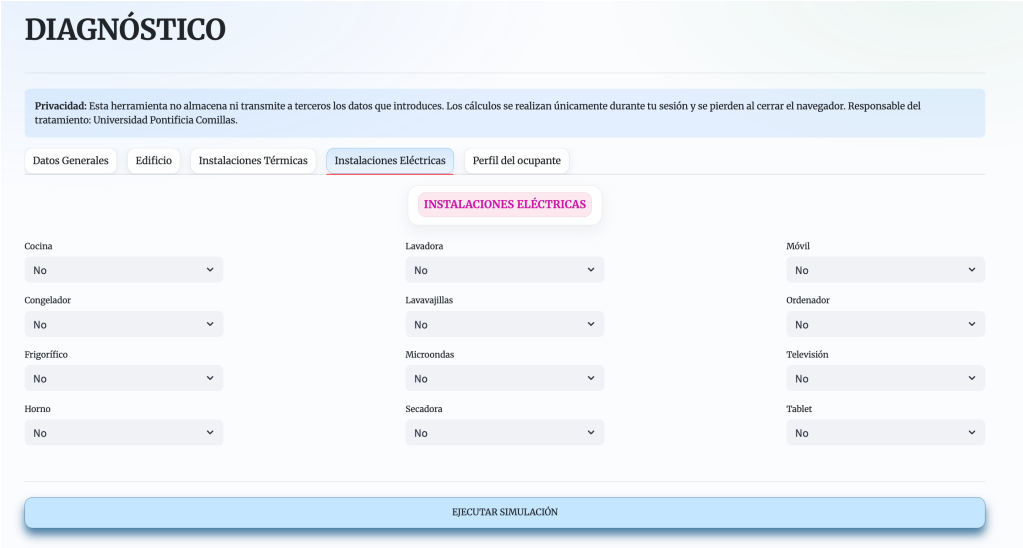


Figura 3.4: Pestaña de Instalaciones Eléctricas.

- **Perfil del ocupante**: recoge el número de personas del hogar según su situación laboral: trabajo activo, desempleo, estudiantes, jubilación, discapacidad o dependencia, viudedad, tareas del hogar y otros. Esta información se utiliza para estimar el consumo eléctrico no térmico, ya que la presencia en el hogar durante el día influye de forma notable en el consumo de iluminación y electrodomésticos.

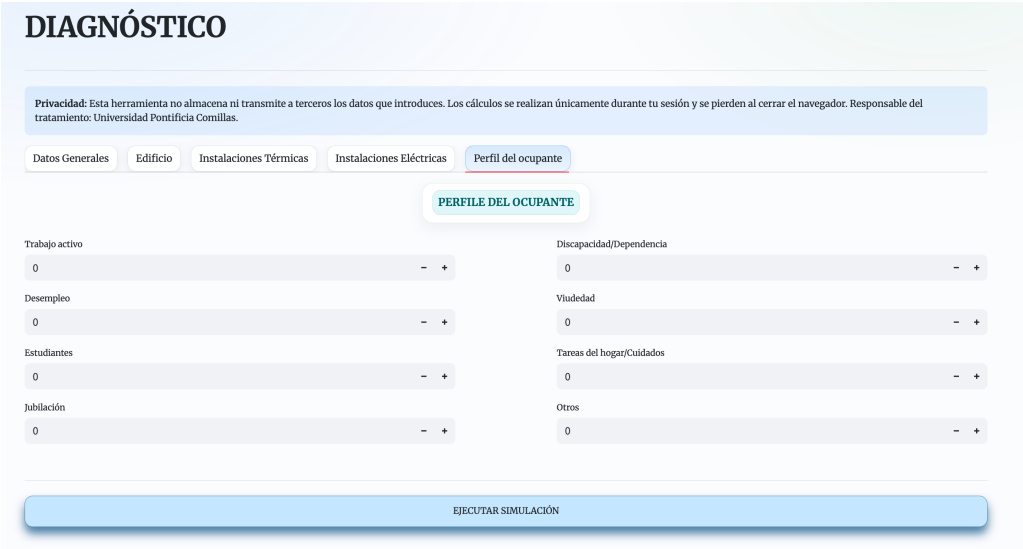


Figura 3.5: Pestaña de Perfil del ocupante.

Además, se decidió mostrar los resultados en una página independiente a la del formulario. Esta decisión también facilitó la organización de la interfaz, ya que permitió dedicar toda la pantalla a presentar los resultados de forma clara y estructurada. Esto resuelve una de las limitaciones del modelo original en EES, donde los resultados y los parámetros de entrada aparecían en el mismo entorno, lo que dificultaba la lectura.

La página de resultados recupera los valores almacenados en `session_state` y los presenta en tres columnas: demanda energética anual (calefacción, refrigeración, ACS y electricidad no térmica), consumo anual (calefacción, ACS, refrigeración y electricidad no térmica) y gasto anual, que incluye el desglose por sistema térmico, el gasto en combustibles y cuatro variantes del gasto eléctrico con y sin refrigeración.

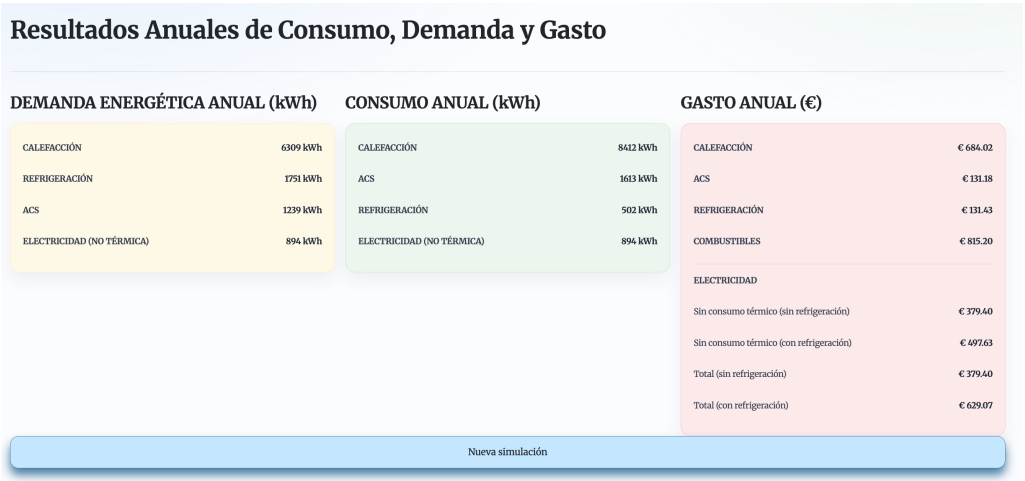


Figura 3.6: Pestaña de resultados finales.

Streamlit

Streamlit es una librería de Python diseñada para construir aplicaciones de datos de forma rápida y sencilla. Su principal ventaja frente a otras opciones es que toda la lógica de la aplicación, tanto el procesamiento de datos como la presentación, se escribe en un único fichero Python. Cada vez que el usuario interactúa con algún elemento de la interfaz, Streamlit re-ejecuta el script de arriba a abajo y actualiza la pantalla con los nuevos valores, lo que simplifica considerablemente la gestión del estado de la aplicación.

En este proyecto, Streamlit se utilizó para construir la interfaz del usuario. Los componentes principales que ofrece la librería y que se han empleado son los siguientes:

- `st.tabs()`: permite organizar el contenido en pestañas. Se utilizó para estructurar las cinco secciones del formulario, de forma que cada pestaña agrupa los campos relacionados con una misma categoría de datos.

- `st.selectbox()`: genera un desplegable con una lista de opciones predefinidas. Se empleó para campos como la provincia, el tipo de vivienda, el sistema de calefacción o el estado de cada electrodoméstico.
- `st.number_input()`: genera un campo numérico con botones de incremento y decremento. Se utilizó para la superficie, la altitud, el número de habitantes y los contadores de ocupación.
- `st.columns()`: permite dividir el espacio horizontal en columnas. Se utilizó para gestionar la disposición de los campos de entrada dentro de cada pestaña.
- `st.button()`: genera un botón interactivo, en este caso "Ejecutar Simulación", que desencadena el cálculo cuando el usuario ha terminado de rellenar los campos de entrada.
- `st.session_state`: Streamlit ejecuta el script completo en cada interacción del usuario con la interfaz, por lo que sin un sistema de almacenamiento, los valores introducidos se perderían en cada interacción. `session_state` es un diccionario asociado a la sesión activa del usuario que permite conservar variables entre ejecuciones consecutivas. Cada campo del formulario queda vinculado a una entrada de `session_state` mediante una clave, lo que garantiza que todos los parámetros introducidos estén disponibles en el momento de llamar al motor de cálculo.
- `st.query_params`: permite leer y escribir parámetros en la URL. Se utilizó para implementar la navegación entre la página de entrada de los datos y la página de resultados sin necesidad de un sistema externo.

Por defecto, Streamlit ofrece un estilo visual fijo con pocas opciones de personalización, por lo que para obtener una interfaz más cuidada se optó por inyectar código CSS directamente en la aplicación.

CSS

El estilo visual de la aplicación se definió íntegramente mediante CSS, con el objetivo de obtener una interfaz más estética y coherente. El código CSS se organiza en varios bloques, cada uno dirigido a un conjunto de elementos concretos.

La tipografía base de la aplicación es *Merriweather*, cargada desde Google Fonts, que se aplica de forma global a todos los elementos de texto. Los títulos utilizan un peso mayor y un color azul oscuro, lo que establece una jerarquía visual clara entre los distintos niveles de contenido. El fondo de la aplicación incorpora un degradado radial sutil que aporta profundidad sin distraer la atención del formulario.

Las pestañas se estilizaron con bordes redondeados, sombra y un fondo diferenciado cuando están seleccionadas, de forma que el usuario pueda identificar fácilmente

en qué sección se encuentra. Los campos de entrada, tanto los desplegables como los numéricos, tienen un tamaño de fuente ligeramente mayor para mejorar la legibilidad.

Dentro de cada pestaña, los grupos de campos se agrupan en paneles con fondo blanco, borde suave y sombra, identificados por una etiqueta de color distinto según su categoría. Esta codificación por colores ayuda a distinguir visualmente los distintos bloques de información sin necesidad de separadores adicionales.

La página de resultados sigue el mismo esquema visual. Cada uno de los tres bloques de resultados (demanda, gasto y consumo) tiene un color de fondo diferente, lo que permite mejor identificación.

Lógica

La secuencia de cálculo que sigue `run_demo()` replica el flujo del modelo original descrito en el capítulo anterior (Figura 2.3).

El flujo de la aplicación se divide en dos páginas: la página del formulario y la página de resultados. La navegación entre ambas se gestiona mediante parámetros en la URL, de forma que cada página tiene una dirección propia y el usuario puede volver al formulario desde los resultados sin perder la sesión. Para evitar que se acceda a páginas no existentes manipulando la URL manualmente, se definió una lista de páginas válidas, de forma que cualquier valor no reconocido redirige automáticamente al formulario.

Dentro de la página del formulario, cada campo de entrada queda registrado en `session_state` con una clave única en el momento en que el usuario introduce un valor. Cuando se pulsa el botón "Ejecutar Simulación", la aplicación recorre todas esas claves, construye un diccionario con los valores recogidos y lo pasa al programa principal. Si el cálculo finaliza correctamente, los resultados se almacenan también en `session_state` y la aplicación redirige a la página de resultados. Si se produce algún error, se muestra un mensaje al usuario indicando que los datos introducidos no han permitido completar el cálculo.

La página de resultados recupera los valores almacenados en `session_state` y los presenta en tres columnas: demanda energética anual, gasto anual en euros y consumo anual, cada una desglosada por sistema.

3.3.2. Acceso a datos

La siguiente fase de implementación fue el desarrollo del módulo de acceso a datos, que constituye la base sobre la que se apoya todo el cálculo energético del modelo.

Una de las diferencias más importantes entre EES y Python en cuanto a la gestión de datos es que EES dispone de un sistema de tablas interno al que se

accede directamente mediante funciones propias del entorno. En Python no existe ese sistema, por lo que fue necesario construirlo desde cero. El módulo `help.py` cumple esa función: centraliza toda la lógica de carga y consulta de las tablas de referencia.

Las tablas se exportaron desde EES a ficheros Excel independientes. En total se cargaron más de 25 tablas, que contienen los datos de referencia necesarios para el cálculo. Incluyen desde las zonas climáticas por provincia y altitud hasta los coeficientes de eficiencia de los sistemas térmicos, las tarifas de suministro por combustible o los pesos de la tarifa eléctrica por tramo horario.

El acceso a los datos se realiza a través de dos funciones:

- `lookup_row()`: dado el nombre de una tabla, una columna y un valor de búsqueda, devuelve el índice de la fila que contiene ese valor. Es el equivalente directo de *Lookup$Row* en EES.
- `lookup_value()`: dado el nombre de una tabla, un índice de fila y un nombre de columna, devuelve el valor de esa celda. Es el equivalente directo de *Lookup* en EES.

Estas dos funciones son las únicas que `model.py` utiliza para acceder a cualquier dato del modelo.

Además de las dos funciones principales de consulta, este módulo incluye también la función `load_all_tables()`, que se encarga de cargar todos los ficheros Excel en memoria al inicio del programa y almacenarlos en un diccionario global accesible desde cualquier punto del código. Cargar todas las tablas al arrancar evita operaciones de lectura repetidas y hace el código más limpio y eficiente.

Un aspecto que generó varios errores durante el desarrollo fue la conversión de tipos en la función `lookup_value()`. Los valores leídos desde los ficheros Excel llegan con el tipo que pandas infiere automáticamente al cargar cada columna: entero, flotante o cadena de texto según el contenido y el formato de las celdas. Esa inferencia no siempre coincide con el tipo que espera la función que va a usar el valor, lo que en algunos casos provocaba comparaciones fallidas o errores de conversión. Por lo tanto, fue necesario añadir lógica de conversión explícita dentro de `lookup_value()` para asegurarse de que el valor devuelto siempre coincide con el tipo que la función espera.

3.3.3. Lógica del cálculo energético

Una vez disponible el módulo de acceso a datos, la segunda fase de implementación fue la traducción de toda la lógica de cálculo del modelo EES a Python. `model.py`, es el fichero central del proyecto que replica función a función el comportamiento de la herramienta DIAGNÓSTICO descrita en el capítulo anterior.

Cabe destacar que dicho módulo es, en esencia, una traducción directa del modelo EES función a función. No se modificó la lógica de cálculo ni las ecuaciones, únicamente se adaptó la sintaxis y la estructur del código para adaptarse a las características de Python; las estructuras de control propias de EES se tradujeron a sus equivalentes naturales en Python, y los nombres de las funciones se mantuvieron lo más parecidos posible a los originales para facilitar la trazabilidad entre ambas versiones.

El módulo se organiza en dos bloques principales que se corresponden con los módulos del modelo original: el bloque térmico y el bloque de electricidad no térmica.

Bloque térmico

El bloque térmico agrupa las funciones encargadas de calcular la demanda, el consumo y el gasto de calefacción, refrigeración y ACS.

La función `zona_climatica()` determina la zona climática de una localidad a partir de su provincia y altitud, devolviendo un código de letra y número. A partir de ella, `zonainv()` y `zonaver()` extraen respectivamente la zona de invierno y la de verano, que son las que utilizan el resto de funciones.

Las demandas de calefacción y refrigeración se calculan mediante las funciones `demanda_referenciacalefaccion()` y `demanda_referenciarefrig()`. Estas consultan las tablas normativas para obtener la demanda de referencia según la zona climática y el tipo de vivienda. `calcula_c1()` y `calcula_c1_verano()` obtienen el índice de calificación energética del edificio, y `io_is()` e `io_is_verano()` calculan el factor de corrección f_c a partir de ese índice y la dispersión de la zona. La función `fdemanda_corregidacal()` aplica una corrección adicional para evitar demandas negativas en zonas sin necesidad de calefacción o refrigeración.

Para el ACS, `elevacion()` calcula la diferencia de altitud entre la localidad y su capital de provincia, que se usa para corregir la temperatura mensual del agua de red antes de calcular la demanda según la ecuación 2.7.

El consumo de cada sistema se obtiene dividiendo la demanda entre su rendimiento estacional. `hspf()` devuelve el SPF del sistema de calefacción según el tipo de instalación y la antigüedad del edificio; `seer()` hace lo propio para refrigeración; y `tabla_acsspf()` selecciona la tabla de rendimientos adecuada para el ACS.

El gasto se calcula mediante un conjunto de funciones que aplican las tarifas, impuestos y el IVA correspondientes según el tipo de combustible y la provincia. La más compleja de este grupo es `tarifa_suministro()`, que gestiona de forma unificada tanto los combustibles no eléctricos como la electricidad, aplicando en cada caso la tarifa correspondiente.

Todas las funciones del bloque siguen el mismo patrón: reciben sus parámetros de entrada, consultan los valores necesarios en las tablas mediante `help.py` y devuelven

un único resultado, sin modificar variables externas ni depender de estado global.

Bloque de electricidad no térmica

El bloque de electricidad no térmica implementa la lógica del módulo eléctrico y se organiza en torno a la función `consumo_electrico()`.

Para cada aparato, la función `consumo_aparato()` determina qué valor de consumo aplicar en función de la respuesta del usuario. Si el aparato está presente, se llama a `gesin()`, que devuelve el consumo individual del aparato para ese tamaño de hogar. Para la mayoría de aparatos este valor se obtiene directamente de la tabla *CMESin*; sin embargo, para aquellos que no están presentes en todos los hogares (lavavajillas, secadora, congelador y ordenador), el valor de consumo medio de la tabla *CMECon* ya incorpora el efecto de la tasa de penetración, por lo que `gesin()` lo divide entre dicha tasa para obtener el consumo real del aparato asumiendo que está presente. Si el aparato no está, el consumo es cero. Si se desconoce, se llama a `gecon()`, que aplica directamente el valor de *CMECon*, que ya incorpora la ponderación por penetración estadística del aparato en hogares de características similares.

El consumo de iluminación se obtiene mediante `gei()`, que consulta la tabla correspondiente a la zona geográfica de la provincia, el número de ocupantes y la superficie de la vivienda.

El ajuste por ocupación se calcula mediante `factores()`, que suma los factores individuales de cada miembro del hogar según su situación laboral. Estos factores, obtenidos de la tabla *Factores*, reflejan la desviación del consumo respecto al hogar medio de igual tamaño en función del tiempo que cada ocupante pasa en la vivienda, tal como se describe en la ecuación 2.15.

El consumo total resultante se multiplica por un factor corrector de 1,07 para tener en cuenta pérdidas y consumos auxiliares no contabilizados de forma explícita.

3.3.4. Coordinación del flujo

Una vez el motor de cálculo y el módulo de acceso a datos quedaron implementados, la tercera y última fase del desarrollo consistió en coordinar el flujo de datos entre la interfaz y el modelo. El módulo `main.py` actúa como agente intermediario entre `app.py` y `model.py` sin contener lógica de cálculo ni elementos visuales.

Su función principal, `run_demo()`, recibe el diccionario de inputs construido por `app.py` y comienza cargando las tablas de referencia en memoria. A continuación extrae y convierte cada parámetro al tipo que corresponde antes de pasarlo a las funciones de cálculo. Este paso es necesario porque los valores recogidos desde la interfaz llegan como cadenas de texto (string) y el modelo requiere tipos concretos en cada caso.

Una vez preparados los inputs, `run_demo()` llama a las funciones de `model.py`. Como se explicó en la sección de planteamiento, Python ejecuta el código de forma secuencial, por lo que es responsabilidad de este módulo garantizar que cada función recibe sus parámetros cuando ya están disponibles. En la práctica, la secuencia de cálculo sigue el siguiente orden: primero se determinan las zonas climáticas de invierno y verano, a continuación se calcula la demanda de ACS, que depende únicamente de la localización y el número de ocupantes, después se calculan las demandas de calefacción y refrigeración, que dependen además de la calificación energética del edificio, luego los consumos de cada sistema, dividiendo cada demanda entre el rendimiento estacional correspondiente, y finalmente los gastos, aplicando las tarifas e impuestos según el tipo de suministro y la provincia.

El consumo eléctrico no térmico se calcula de forma independiente a partir del equipamiento del hogar y la ocupación de sus miembros.

Al finalizar, los resultados se empaquetan en un diccionario con tres grupos de salidas: demandas energéticas anuales (calefacción, refrigeración, ACS y electricidad no térmica), consumos anuales (calefacción, ACS, refrigeración y electricidad no térmica) y gastos anuales desglosados por sistema (calefacción, ACS, refrigeración y combustibles). A estos se añaden cuatro salidas específicas del gasto eléctrico: el gasto sin consumo térmico sin refrigeración, el gasto sin consumo térmico con refrigeración, el gasto eléctrico total sin refrigeración y el gasto eléctrico total con refrigeración. Este diccionario es el que `app.py` recibe y presenta al usuario en la página de resultados.

La siguiente figura muestra de forma esquemática cómo se relacionan los cuatro módulos del sistema.

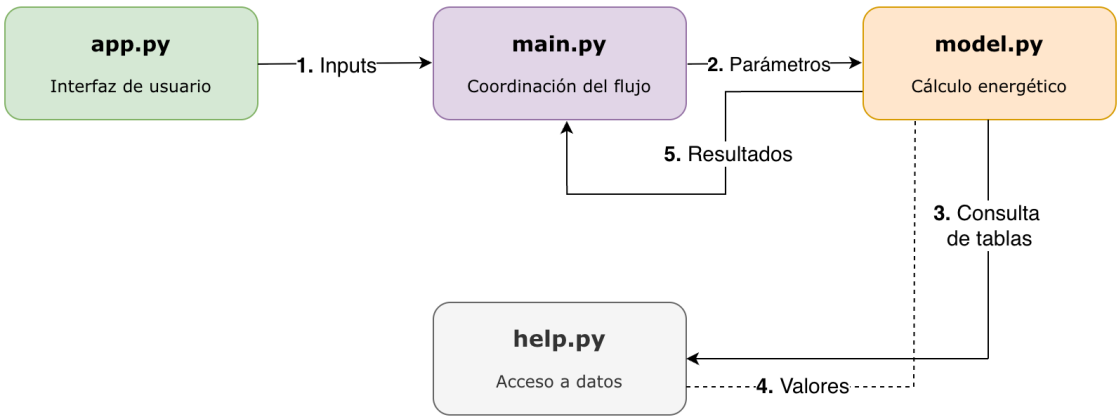


Figura 3.7: Diagrama de relaciones entre los módulos del sistema.

3.4. AUDITORÍA DE SEGURIDAD

Una vez completada la implementación de los cuatro módulos, se llevó a cabo una verificación del funcionamiento del modelo. Se ejecutaron pruebas con distintas combinaciones de entradas, comparando los resultados obtenidos en Python con los del modelo original en EES. Los valores de demanda, consumo y gasto coincidieron en todos los casos comprobados, lo que confirmó que la traducción era fiel al modelo original.

Con el modelo validado, el siguiente paso fue estudiar la viabilidad de convertir la aplicación en una herramienta accesible desde la web. Para ello, se realizó una auditoría de seguridad con el objetivo de identificar posibles vulnerabilidades y garantizar que la aplicación cumplía con los requisitos mínimos de seguridad y privacidad.

3.4.1. Dependencias y configuración

Se generó un fichero `requirements.txt` con las versiones exactas de las dependencias del proyecto (Streamlit, pandas, openpyxl y NumPy) y se verificó que ninguna de ellas presentaba vulnerabilidades conocidas mediante la herramienta `pip-audit`. Fijar las versiones es una práctica recomendada ya que sin ella, una actualización automática de una librería podría introducir cambios inesperados en el comportamiento de la aplicación o, en el peor caso, incorporar una vulnerabilidad de seguridad conocida. Usando pip-audit, se contrasta cada dependencia contra bases de datos públicas de vulnerabilidades, de forma que si alguna versión tiene un problema de seguridad documentado se detecta antes del despliegue.

Además, se creó el fichero `.streamlit/config.toml` con tres opciones de configuración, entre ellos `showErrorDetails = false`, para evitar que los mensajes de error internos se muestren en el navegador y `enableXsrfProtection = true`, que activa la protección frente a ataques de tipo *cross-site request forgery*. Este consiste en un tipo de ataque en el que una página web maliciosa puede enviar peticiones a la aplicación en nombre del usuario sin que este lo sepa. Sin esta protección, un tercero podría interactuar con la aplicación suplantando al usuario legítimo. Asimismo, exponer registros de error al usuario final es un riesgo elevado en aplicaciones web ya que un mensaje de error detallado puede revelar rutas internas del servidor, nombres de ficheros o fragmentos de código que faciliten a un atacante identificar puntos débiles del sistema.

3.4.2. Control de errores

La llamada a `run_demo()` se envolvió en un bloque `try/except` que distingue entre dos tipos de error. El primero es un error de validación (`ValueError`), que ocurre cuando la combinación de parámetros introducida no es válida para el modelo.

El segundo es cualquier error inesperado. En ambos casos el usuario recibe un mensaje genérico avisando de un fallo, mientras que el error interno queda registrado en el sistema mediante el uso de `logging`.

Esta separación tiene dos propósitos. Por un lado, protege la aplicación frente a usuarios malintencionados: si un error expusiera información interna como rutas de ficheros o fragmentos del modelo, podría ser aprovechada para explotar vulnerabilidades. Por otro, el registro mediante `logging` permite al administrador revisar qué ha fallado y en qué condiciones, sin que esa información sea visible desde el exterior.

3.4.3. Navegación y privacidad

Por último, se añadió un aviso de privacidad visible en la página principal de la aplicación. Este aviso informa al usuario de que los datos introducidos no se almacenan ni se transmiten a terceros, y que los cálculos se realizan únicamente durante la sesión activa. Este aviso responde a los requisitos del Reglamento General de Protección de Datos (RGPD), que obliga a informar a los usuarios sobre el tratamiento de sus datos personales antes de que los faciliten. El texto definitivo, con el enlace a la política institucional de Comillas, quedará pendiente de confirmación por parte del DPO de la universidad.

Por último, se creó un fichero `.gitignore` para excluir del repositorio los ficheros generados automáticamente como `__pycache__`, `.DS_Store` o temporales de Excel, y se eliminaron los que ya habían sido incluidos. Incluir este tipo de ficheros en un repositorio no solo añade ruido innecesario, sino que en algunos casos puede exponer información sensible del entorno de desarrollo, como rutas absolutas del sistema o metadatos del sistema operativo.

Capítulo 4. Comparación de modelos y análisis de resultados

4.1. METODOLOGÍA DE VALIDACIÓN

Para comprobar la implementación del modelo en Python, se ha llevado a cabo una validación sistemática comparando los resultados obtenidos con los del modelo de referencia en EES.

La validación se ha basado en 25 casos de prueba, diseñados de forma que el conjunto cubra al menos una vez cada una de las opciones posibles de las variables de entrada. En concreto, se ha garantizado la presencia de:

- **Todos los sistemas de calefacción disponibles:** gas natural, gasóleo, GLP, biomasa, carbón, electricidad por radiadores, electricidad por acumuladores y bomba de calor
- **Todos los sistemas de ACS:** gas natural, GLP, biomasa, carbón y electricidad
- **Los tres tipos de instalación para calefacción:** individual, central y por aparatos; y **los dos para ACS:** individual y central
- **Ambos tipos de vivienda:** bloque y unifamiliar.
- **Todas las calificaciones energéticas:** desde Anterior a 1981 hasta G
- **Todos los perfiles de ocupante**
- **Provincias de distintas zonas climáticas**
- **Distintas altitudes:** (capitales de provincia y altitudes intermedias)

Para ejecutar la validación, se desarrolló un script en Python que llama a `run_demo()` de forma automática para cada uno de los 25 casos, sin intervención manual entre ejecuciones. Esto permitió obtener los 25 conjuntos de resultados en una única ejecución y exportarlos a un fichero Excel para su comparación con los valores obtenidos manualmente en EES. Este proceso ilustra además una de las ventajas prácticas de la migración; mientras que en EES cada caso requería introducir los parámetros y ejecutar el modelo de forma individual, en Python el mismo proceso puede automatizarse completamente.

Para cada caso, se han introducido los mismos datos de entrada en ambos modelos y se han comparado las 16 variables de salida del modelo: demanda, consumo y

gasto de calefacción, refrigeración, ACS y electricidad no térmica, más el gasto en combustibles y el gasto eléctrico total con y sin refrigeración.

Los 25 casos de prueba se recogen en la Tabla 4.1. Con el fin de no sobrecargar la tabla, se omiten las columnas de electrodomésticos y perfil del ocupante, que pueden consultarse en el anexo 5.4.

Tabla 4.1: Casos de validación empleados en la comparación Python–EES

Provincia	**Calif.**	**Tipo**	**Calefacción**	**Inst.**	**ACS**	**Inst.**
Madrid	Ant. 1981	B	Gas Natural	Indiv.	Gas Natural	Indiv.
Barcelona	1981–2007	U	Gas Natural	Central	Gas Natural	Central
Sevilla	Post. 2007	B	Bomba de calor	Indiv.	Electricidad	Indiv.
Zaragoza	A	U	Elec. (radiadores)	Indiv.	Gas Natural	Indiv.
Bizkaia	B	B	Elec. (acumul.)	Indiv.	Electricidad	Central
Valencia	C	U	Gasóleo	Indiv.	Gas Natural	Indiv.
Málaga	D alta	B	GLP	Indiv.	GLP	Indiv.
Burgos	D baja	U	Biomasa	Indiv.	Biomasa	Indiv.
Granada	E alta	B	Carbón	Indiv.	Carbón	Indiv.
Coruña, A	E media	U	Gas Natural	Central	Gas Natural	Central
Alicante	E baja	B	Bomba de calor	Indiv.	Electricidad	Indiv.
Huelva	F	U	Gas Natural	Indiv.	Biomasa	Indiv.
Ceuta	G	B	GLP	Indiv.	GLP	Indiv.
Melilla	Ant. 1981	U	Gasóleo	Indiv.	Electricidad	Indiv.
Salamanca	1981–2007	B	Elec. (radiadores)	Apar.	Gas Natural	Indiv.
Lleida	C	U	Gas Natural	Indiv.	Carbón	Indiv.
Murcia	D alta	B	Bomba de calor	Indiv.	Gas Natural	Central
Huesca	E alta	U	Biomasa	Central	Gas Natural	Indiv.
Almería	F	B	Elec. (acumul.)	Apar.	Electricidad	Indiv.
Pontevedra	G	U	Gas Natural	Central	GLP	Indiv.
Toledo	B	B	Carbón	Indiv.	Gas Natural	Indiv.
Cádiz	A	U	Elec. (radiadores)	Indiv.	Electricidad	Indiv.
Teruel	1981–2007	B	Gasóleo	Central	Biomasa	Indiv.
Córdoba	D baja	U	Bomba de calor	Apar.	Gas Natural	Indiv.
Asturias	E media	B	Gas Natural	Central	Electricidad	Central

Indiv. = individual; Apar. = aparatos; B = Bloque; U = Unifamiliar

Para evaluar la precisión del modelo de Python respecto al modelo de EES, se analizan los errores a dos niveles: por variable individual, calculando el error relativo porcentual para cada una de las variables de salida; y por categoría, calculando el error absoluto medio (MAE) de demanda, consumo y gasto respectivamente.

El error relativo porcentual para la observación i se calcula como:

$$\varepsilon_i = \frac{v_{\text{Python},i} - v_{\text{EES},i}}{v_{\text{EES},i}} \times 100 \tag{4.1}$$

y el MAE de cada categoría como:

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\varepsilon_i| \tag{4.2}$$

donde N es el número total de observaciones de esa categoría (25 casos multiplicados por el número de variables del grupo).

4.2. RESULTADOS DE LA VALIDACIÓN

4.2.1. Precisión por variable

La Tabla 4.2 muestra el error relativo porcentual medio, en valor absoluto, para cada salida del modelo.

Tabla 4.2: Error relativo porcentual medio por variable de salida

Variable	$\overline{\lvert\varepsilon\rvert}$ (%)	$\lvert\varepsilon\rvert_{\text{máx}}$ (%)
Demanda energética (kWh)		
Calefacción	0,0087	0,0358
Refrigeración	0,0053	0,0256
ACS	0,0155	0,0356
Electricidad no térmica	0,0085	0,0192
Consumo (kWh)		
Calefacción	0,0070	0,0319
ACS	0,0122	0,0299
Refrigeración	0,0254	0,0953
Electricidad no térmica	0,0085	0,0192
Gasto (€)		
Calefacción	0,0063	0,0381
ACS	0,0023	0,0165
Refrigeración	0,0001	0,0030
Combustibles	0,0032	0,0202
Electricidad no térmica (sin ref.)	0,0050	0,0298
Electricidad no térmica (con ref.)	0,0054	0,0298
Electricidad total (sin ref.)	0,0048	0,0298
Electricidad total (con ref.)	0,0048	0,0298

En todos los casos los errores medios quedan por debajo del 0,03 % y los máximos individuales no superan el 0,10 %. La variable con mayor error medio es el consumo de refrigeración (0,0254 %), seguida de la demanda de ACS (0,0155 %). Los valores máximos se concentran en casos concretos y no apuntan a ningún error sistemático.

El error máximo de consumo de refrigeración (0,0953 %) corresponde al caso 24 (Córdoba, calificación D baja, bomba de calor), el cual coincide con la demanda de refrigeración más alta del conjunto (3 448 kWh). Sin embargo, no se ha identificado ninguna causa estructural que explique que el error sea mayor en este caso concreto, y además, no se repite en ninguna otra variable del mismo caso.

4.2.2. Precisión agregada por categoría

La Tabla 4.3 resume la precisión global agrupando todas las observaciones de cada categoría.

Tabla 4.3: Error absoluto medio (MAE) agregado por categoría

Categoría	**MAE (%)**
Demanda energética	0,0095
Consumo	0,0133
Gasto	0,0049

El MAE más alto corresponde al consumo (0,0133 %), seguido de la demanda (0,0095 %) y el gasto (0,0049 %). Esta jerarquía tiene una explicación directa en la estructura del modelo: el consumo se obtiene dividiendo la demanda por el rendimiento estacional (SPF o SEER), de manera que cualquier diferencia de redondeo en la demanda se traslada al paso siguiente. El gasto, en cambio, resulta de multiplicar el consumo por tarifas comunes a ambos modelos, lo que tiende a compensar las diferencias relativas acumuladas en cálculos anteriores.

4.3. ANÁLISIS DE RESULTADOS

4.3.1. Interpretación del error residual

Tras el análisis de los errores obtenidos, se ha determinado que las diferencias tienen un único origen: los redondeos acumulados en los pasos intermedios del cálculo. EES y Python no gestionan la precisión numérica de forma idéntica, lo que produce pequeñas discrepancias en los resultados intermedios que se propagan a lo largo de la cadena de cálculo hasta aparecer en las salidas finales. En ningún caso se ha identificado un error de lógica en la implementación: los errores no siguen ningún patrón reconocible por provincia, sistema de calefacción ni calificación energética, lo que confirma que las diferencias son atribuibles exclusivamente a la precisión numérica interna de cada entorno.

4.3.2. Distribución de errores por zona climática

Aunque los errores medios son bajos en todos los casos, tiene sentido preguntarse si los casos en zonas climáticas extremas tienden a concentrar errores más altos. El consumo de refrigeración es la variable con mayor error medio del conjunto (0,0254 %), por lo que se analizó su distribución agrupando los 25 casos según su zona climática de verano.

La Tabla 4.4 recoge la zona climática de invierno y verano de cada caso.

Tabla 4.4: Zona climática de invierno y verano de los 25 casos de validación

Caso	Provincia	Altitud (m)	ZCI	ZCV
1	Madrid	8000	D	3
2	Barcelona	8000	C	2
3	Sevilla	8000	B	4
4	Zaragoza	500	D	3
5	Bizkaia	8000	C	1
6	Valencia	8000	B	3
7	Málaga	8000	A	3
8	Burgos	900	E	1
9	Granada	700	C	3
10	A Coruña	8000	C	1
11	Alicante	8000	B	4
12	Huelva	8000	A	4
13	Ceuta	8000	B	3
14	Melilla	8000	A	3
15	Salamanca	800	D	2
16	Lleida	1000	E	1
17	Murcia	8000	B	3
18	Huesca	1200	E	1
19	Almería	8000	A	4
20	Pontevedra	400	D	1
21	Toledo	600	D	3
22	Cádiz	8000	A	3
23	Teruel	1100	E	1
24	Córdoba	8000	B	4
25	Asturias	300	D	1

ZCI: zona climática de invierno; ZCV: zona climática de verano

Los casos se distribuyen de la siguiente forma:

- **Zona 1:** Bizkaia (5), Burgos (8), A Coruña (10), Lleida (16), Huesca (18), Pontevedra (20) y Asturias (25).
- **Zona 2:** Barcelona (2) y Salamanca (15).

- **Zona 3:** Zaragoza (4), Valencia (6), Málaga (7), Granada (9), Ceuta (13), Melilla (14), Murcia (17), Toledo (21) y Cádiz (22).
- **Zona 4:** Madrid (1), Sevilla (3), Alicante (11), Huelva (12), Almería (19), Teruel (23) y Córdoba (24).

Los casos de zona 1 tienen consumo de refrigeración nulo por definición, ya que el modelo asigna demanda cero a las localidades con zona climática de verano igual a 1. Su error es por tanto cero en todos los casos y no aportan información sobre el comportamiento numérico del modelo en este módulo.

En el resto de los casos se presentan los siguientes errores relativos medios de consumo de refrigeración por zona climática de verano:

Tabla 4.5: Error relativo medio del consumo de refrigeración por zona climática de verano

Zona climática de verano	**Error medio (%)**
1	0,0000
2	0,0603
3	0,0370
4	0,0226

La distribución completa de casos por zona puede consultarse en el Anexo III.

En el resto de zonas, el error no sigue ninguna tendencia clara: zona 2 presenta el error medio más alto (0,0603 %), mientras que zona 4, donde las demandas de refrigeración son más altas, tiene el más bajo (0,0226 %). Si el error dependiera de la magnitud de la demanda, se esperaría lo contrario. El valor máximo individual es el del caso 24 (Córdoba, zona 4), con un 0,0953 %, pero ese mismo caso no presenta errores elevados en el resto de variables, lo que descarta un fallo puntual en su cálculo.

En conjunto, los resultados no muestran ningún patrón asociado a la zona climática, lo que refuerza la conclusión de que las diferencias entre Python y EES son atribuibles exclusivamente al redondeo numérico.

4.3.3. Patrones en el gasto eléctrico

Como se describió en la sección 2.3.4, el modelo calcula el gasto eléctrico no térmico en dos variantes: con y sin refrigeración. La diferencia entre ambas no es el coste de la refrigeración en sí, sino el sobrecoste del término fijo al contratar más potencia. En los casos en que el sistema de calefacción ya obliga a contratar la potencia máxima (radiadores eléctricos o bomba de calor con demanda suficientemente alta), añadir refrigeración no altera la potencia contratada y, por tanto, `noterm_sinREF` = `noterm_conREF`. Sus errores son por tanto idénticos, como se puede comprobar

en los casos 4 (Zaragoza), 15 (Salamanca), 22 (Cádiz), 3 (Sevilla), 17 (Murcia) y 24 (Córdoba). En el resto de casos con refrigeración, `noterm_conREF` es mayor que `noterm_sinREF` porque la potencia contratada sube de 5,0 a 6,9 kW.

Este patrón se refleja también en los errores máximos de la Tabla 4.2. Las cuatro variables de gasto eléctrico comparten el mismo valor máximo (0,0298 %), lo que indica que el caso con mayor error es precisamente uno en que `noterm_sinREF` = `noterm_conREF` y, por tanto, los errores de las cuatro salidas coinciden.

Cuando calefacción y ACS funcionan con combustible, no hay gasto eléctrico de climatización y `Elec. total sinREF` coincide con `noterm_sinREF`, por lo que sus errores son también idénticos. Es el caso de Madrid (gas natural + gas natural), Valencia (gasóleo + gas natural) o Málaga (GLP + GLP), entre otros.

4.3.4. Provincias con mayor error por variable

La Tabla 4.6 recoge, para cada variable de salida, la provincia con mayor error relativo porcentual individual.

Tabla 4.6: Error relativo máximo individual por variable de salida

Variable	**Error máx. (%)**	**Provincia**
Demanda (kWh)		
Calefacción	0,0358	Córdoba
Refrigeración	0,0256	Valencia
ACS	0,0356	Barcelona
Electricidad no térmica	0,0192	Melilla
Consumo (kWh)		
Calefacción	0,0319	Murcia
ACS	0,0299	Ceuta
Refrigeración	0,0953	Córdoba
Electricidad no térmica	0,0192	Melilla
Gasto (€)		
Calefacción	0,0381	Murcia
ACS	0,0165	Barcelona
Refrigeración	0,0030	Melilla
Combustibles	0,0202	Burgos
Electricidad no térmica (sinREF)	0,0298	Burgos
Electricidad no térmica (conREF)	0,0298	Burgos
Electricidad total (sinREF)	0,0298	Burgos
Electricidad total (conREF)	0,0298	Burgos

Los errores máximos se reparten entre casos distintos: no hay ningún caso que concentre los errores más altos en todas las variables, lo que descarta cualquier fallo

sistemático. Burgos acumula los cuatro errores máximos del gasto eléctrico, pero todos corresponden a la misma diferencia propagada a las cuatro variantes de esa variable, por lo que en la práctica es un único error.

Murcia presenta el error máximo en gasto y consumo de calefacción, y Córdoba en demanda y consumo de refrigeración. A pesar de ello, ese error no aparece en el resto de variables del mismo caso, lo que descarta que haya algún atributo o valor de entrada que lo haga más propenso al error. Si hubiera un fallo de lógica en alguna función, todos los casos que pasan por esa misma rama del cálculo mostrarían un error similar y reconocible, algo que no ocurre en ningún caso del conjunto. La ausencia de cualquier patrón reconocible confirma que las diferencias son atribuibles exclusivamente al redondeo numérico.

4.3.5. Conclusión

La validación sobre 25 casos de prueba ha confirmado que el modelo en Python reproduce el comportamiento del modelo original en EES con una precisión equivalente a la del modelo original en EES. En ninguna de las 16 variables de salida se ha identificado un error atribuible a un fallo de lógica en la implementación, lo que permite concluir que la traducción función a función ha sido fiel al modelo original.

Desde el punto de vista práctico, los errores obtenidos no tienen ninguna consecuencia sobre el uso de la herramienta. Una diferencia media inferior al 0,02 % en cualquier variable de salida no es determinante ni en la identificación de hogares en situación de vulnerabilidad energética ni en el diagnóstico del gasto energético requerido.

La robustez de estos resultados está respaldada por la diversidad de los casos analizados. Los 25 casos cubren todos los sistemas de calefacción y ACS disponibles, ambos tipos de vivienda, todas las calificaciones energéticas y provincias representativas de todas las zonas climáticas de la península. El hecho de que los errores sean uniformemente bajos y sin ningún patrón reconocible a lo largo de un conjunto tan variado de configuraciones refuerza la validez de la implementación.

Por último, la capacidad de ejecutar el modelo de forma automática sobre múltiples casos es una de las mejoras más concretas que aporta la migración. En EES, cada simulación requería introducir los parámetros manualmente y ejecutar el modelo de forma individual, mientras que en Python, el mismo proceso puede aplicarse de forma sistemática a bases de datos de hogares mucho más grandes. Esto abre la puerta a los análisis a gran escala que constituían uno de los objetivos del proyecto y que no eran viables con la implementación original.

Capítulo 5. Conclusiones y futuros estudios

5.1. CONCLUSIONES SOBRE LA METODOLOGÍA

En términos generales, el proceso de desarrollo ha sido coherente con los objetivos planteados al inicio, aunque con algunas dificultades que no estaban del todo previstas.

El desarrollo del proyecto ha permitido comprobar que migrar un modelo de cálculo energético de EES a Python es viable sin necesidad de modificar la lógica de cálculo ni las ecuaciones del modelo original. La traducción función a función resultó ser un enfoque acertado: al mantener una correspondencia directa entre el código EES y el código Python, fue posible verificar en todo momento que el comportamiento de cada función era el esperado, y localizar errores con relativa facilidad cuando los resultados no coincidían.

La decisión de organizar el código en cuatro módulos con responsabilidades delimitadas también ha demostrado ser correcta. La separación en módulos permitió modificar partes del sistema de forma independiente sin afectar al resto. Esta separación también facilitó la depuración, ya que cuando aparecía un error era más sencillo identificar en qué módulo se producía. De cara al futuro, esta arquitectura hace que el sistema sea más fácil de mantener y de ampliar: añadir una nueva variable de entrada, actualizar una tarifa o cambiar el comportamiento de una función de cálculo son tareas que se pueden hacer de forma localizada sin riesgo de introducir efectos inesperados en otras partes.

Otro aspecto a destacar es la elección de Streamlit como librería para la interfaz de usuario para el alcance del proyecto. Su principal ventaja es que permite construir una aplicación interactiva completa en Python puro, sin necesidad de escribir HTML o JavaScript por separado, ni de gestionar una arquitectura de frontend y backend. Esto redujo considerablemente el tiempo de desarrollo de la interfaz y permitió centrarse en la lógica del modelo. Además, el CSS personalizado permitió obtener una interfaz más cuidada que la que ofrece Streamlit por defecto.

Dicho esto, Streamlit también tiene limitaciones que se hicieron evidentes durante el desarrollo. La más notable fue la gestión del estado entre páginas: Streamlit re-ejecuta el script completo en cada interacción del usuario, lo que hace que los valores introducidos en el formulario se pierdan si no se gestionan correctamente. Esto obligó a replantear la gestión del estado entre páginas, añadiendo complejidad que no estaba prevista al inicio.

Otro aspecto que resultó más costoso de lo esperado fue la preparación de las

tablas de referencia. Aunque la lógica de acceso a datos es conceptualmente sencilla, la exportación de las tablas desde EES introdujo inconsistencias de formato que tuvieron que revisarse manualmente una a una. Este proceso no requería conocimientos técnicos especiales, pero sí mucha atención al detalle y consumió una parte considerable del tiempo de desarrollo.

La auditoría de seguridad, aunque no estaba prevista inicialmente como una fase del proyecto, resultó ser una etapa necesaria y útil. Más allá de los cambios concretos que produjo en el código, el proceso de pensar en la aplicación desde el punto de vista de la seguridad ayudó a identificar decisiones de diseño que podían mejorarse, como la gestión de los mensajes de error o la validación de los parámetros de la URL. Incorporar este tipo de revisión desde las primeras fases del desarrollo, en lugar de hacerla al final, habría permitido tomar algunas de esas decisiones de forma más consciente desde el principio.

Además, una de las ventajas de migrar el modelo a una interfaz web es que ha permitido añadir comprobaciones que en EES no eran posibles. Antes de ejecutar el cálculo, la aplicación verifica que el perfil del ocupante sea coherente con el número de habitantes declarado: si la suma de las categorías no coincide, se muestra un mensaje de error y se detiene la ejecución. Esta comprobación evita que el modelo reciba datos inconsistentes que podrían producir resultados incorrectos sin ningún aviso al usuario.

En conjunto, el proceso de desarrollo ha seguido una progresión lógica y los resultados son coherentes con los objetivos planteados al inicio. Las dificultades encontradas han sido en su mayoría de naturaleza técnica y se han resuelto sin necesidad de replantear el enfoque general. La arquitectura del sistema resultante es modular y clara, lo que permite que futuras mejoras o ampliaciones puedan abordarse sin dificultad.

Las decisiones tomadas durante el desarrollo han permitido cumplir los objetivos planteados al inicio del proyecto. La organización modular facilitó tanto el desarrollo como la verificación, y la elección de Streamlit permitió construir una interfaz funcional sin desviar el foco de la lógica de cálculo. El resultado es una herramienta que replica el comportamiento de DIAGNÓSTICO en un entorno abierto, automatizable e integrable con otras herramientas digitales, que era precisamente lo que la implementación original en EES no permitía.

5.2. LIMITACIONES

La migración del modelo a Python reproduce fielmente la lógica del modelo original, pero el proceso puso de manifiesto algunas diferencias entre ambos entornos que conviene documentar.

La primera tiene que ver con el manejo de tipos. EES no distingue de forma estricta entre texto y número, y sus tablas admiten valores mixtos sin problema. En Python,

pandas puede interpretar una misma columna de formas distintas dependiendo de si hay celdas vacías, valores con formato europeo (coma como separador decimal) o mezcla de tipos. En la práctica esto se tradujo en errores que no eran evidentes hasta ejecutar el modelo con una combinación concreta de entradas: un valor que visualmente parecía un número llegaba al cálculo como cadena de texto, lo que provocaba fallos en las operaciones aritméticas. El problema se resolvió de forma sistemática mejorando la función `lookup_value()`, que ahora intenta convertir a `float` cualquier valor numérico antes de devolverlo. Aun así, es un aspecto a tener en cuenta si en el futuro se actualizan o amplían las tablas de referencia, ya que una modificación en el formato de un fichero Excel podría reintroducir el mismo tipo de error.

Otra diferencia relevante es que EES resuelve todas las ecuaciones de forma simultánea, sin importar el orden en que están definidas, mientras que Python ejecuta el código de forma secuencial. En EES esto no es un problema: el solver encuentra la solución independientemente de cómo esté organizado el código. En Python, en cambio, cada variable debe estar calculada antes de ser utilizada. Esto obligó a analizar con detalle las dependencias entre funciones y a reestructurar el orden de cálculo en `main.py`, de forma que cada función recibiera sus parámetros cuando ya estuvieran disponibles. En general la secuencia resultó bastante natural, pero hubo casos en los que una variable se necesitaba antes de lo que inicialmente se había previsto, lo que requirió ajustar el orden varias veces durante el desarrollo.

La segunda limitación está relacionada con la exportación de las tablas de referencia desde EES. Al exportarlas a ficheros Excel, el formato de algunos valores se alteró: separadores decimales que cambiaron de coma a punto, valores numéricos que se convirtieron en texto, o nombres de columnas con espacios que no eran visibles a simple vista. Estas inconsistencias no generaban errores al cargar las tablas, sino al intentar buscar un valor concreto, lo que las hacía especialmente difíciles de detectar. Cada tabla tuvo que revisarse y corregirse manualmente, lo que supuso una parte significativa del tiempo de desarrollo, para garantizar que los valores que lee el modelo coinciden exactamente con los del original.

Más allá de estas diferencias técnicas, el modelo hereda también las limitaciones del original. Las tarifas eléctricas y de gas, los impuestos y los datos normativos están fijados en las tablas de referencia y no se actualizan automáticamente. Esto significa que los resultados son válidos únicamente para los valores vigentes en el momento de la última actualización de dichas tablas. En un contexto en el que los precios de la energía han experimentado variaciones importantes en los últimos años, esta limitación puede afectar a la precisión de los resultados si las tablas no se mantienen al día.

5.3. CONCLUSIONES SOBRE LOS RESULTADOS

La validación del modelo sobre 25 casos de prueba ha permitido comprobar que la migración de EES a Python es correcta desde el punto de vista numérico. Los errores medios se sitúan por debajo del 0,03 % en todas las variables de salida, y el MAE agregado no supera el 0,014 % en ninguna de las tres categorías analizadas: demanda, consumo y gasto. Estas diferencias son del orden esperable entre dos entornos de cálculo que no gestionan la precisión numérica de forma idéntica, y no tienen ningún efecto práctico sobre los resultados del modelo.

La jerarquía de errores entre categorías responde a la estructura del propio modelo. El consumo presenta el MAE más alto (0,0133 %) porque se obtiene dividiendo la demanda por el rendimiento estacional, lo que amplifica ligeramente cualquier diferencia acumulada en el paso anterior. El gasto, en cambio, tiene el MAE más bajo (0,0049 %) porque resulta de multiplicar el consumo por tarifas que son idénticas en ambos modelos, lo que tiende a compensar las diferencias relativas acumuladas. Esta progresión es coherente con la cadena de cálculo del modelo y no indica ningún problema en la implementación.

El análisis por zona climática de verano tampoco reveló ningún patrón sistemático. El error medio del consumo de refrigeración no crece con la intensidad climática; la zona 4, donde las demandas son más altas, presenta el error medio más bajo de las zonas con refrigeración (0,0226 %), mientras que la zona 2 presenta el más alto (0,0603 %). Esto descarta que el error dependa de la magnitud de la demanda o de las condiciones climáticas del caso analizado.

Los errores máximos individuales tampoco siguen ningún patrón reconocible. Se reparten entre casos distintos y en ninguno de ellos el error máximo en una variable se repite en las demás. Esto es coherente con un origen puramente numérico. Un error de redondeo en un paso intermedio afecta únicamente a las variables que dependen de ese paso y no se propaga a variables que se calculan por ramas o funciones independientes del modelo. Si hubiera un error de lógica en alguna función, aparecería de forma consistente en todos los casos que pasan por esa función, independientemente de sus otras características. Que eso no ocurra en ningún caso es la evidencia más directa de que la traducción función a función ha sido fiel al modelo original.

En conjunto, los resultados de la validación permiten afirmar que el modelo en Python reproduce el comportamiento del modelo original en EES con una precisión equivalente a la del modelo original en EES. La herramienta puede aplicarse con confianza tanto a casos individuales como a análisis masivos sobre bases de datos de hogares, sin que las diferencias numéricas respecto al modelo original tengan ninguna consecuencia práctica sobre las conclusiones que se extraigan.

5.4. RECOMENDACIONES PARA FUTUROS ESTUDIOS

El trabajo desarrollado deja abiertas varias líneas de mejora que no formaban parte del alcance de este proyecto pero que serían continuaciones naturales del mismo.

El más directo sería el despliegue de la aplicación como herramienta web accesible desde un navegador, sin necesidad de instalar Python ni ejecutar el código en local. La arquitectura del proyecto ya está preparada para ello: la separación entre interfaz, motor de cálculo y acceso a datos facilita el despliegue, y la auditoría de seguridad inicial descrita en el capítulo anterior cubre los requisitos mínimos para una primera versión. No obstante, antes del despliegue definitivo sería necesario realizar una auditoría más exhaustiva adaptada al entorno de producción concreto, confirmar los textos legales con el DPO de la universidad, y valorar si la herramienta requiere algún sistema de autenticación de usuarios en función del uso previsto y del perfil de las personas que vayan a utilizarla.

Ampliar el modelo con nuevas variables de entrada sería otra línea de trabajo natural. Uno de los objetivos del proyecto era preparar la herramienta para su aplicación a gran escala, de forma que pudiera ejecutarse de forma masiva sobre bases de datos de hogares. Actualmente el modelo cubre los principales sistemas térmicos y el equipamiento eléctrico más habitual, pero podría enriquecerse con parámetros como la presencia de energía solar térmica o fotovoltaica, sistemas de almacenamiento, o una caracterización más detallada del nivel de aislamiento del edificio. Cada nueva variable requeriría revisar y ampliar las tablas de referencia correspondientes, adaptar la interfaz de usuario y comprobar que los nuevos cálculos son coherentes con el resto del modelo.

Otra mejora útil sería añadir un sistema de exportación de resultados. En la versión actual, cada simulación se ejecuta de forma independiente y los resultados solo son accesibles durante la sesión activa: en cuanto el usuario cierra el navegador o inicia una nueva simulación, los datos anteriores se pierden. Generar automáticamente un fichero con las entradas introducidas y los resultados obtenidos en cada ejecución, organizado por carpetas según criterios como la fecha, la provincia o el perfil del hogar, permitiría conservar un registro actualizado de las simulaciones realizadas. Esto facilitaría tanto la consulta de resultados anteriores sin necesidad de repetir la simulación como el análisis comparativo de múltiples hogares, y abriría la posibilidad de incorporar una vista de historial en la propia interfaz desde la que el usuario pudiera recuperar y comparar ejecuciones previas directamente.

Desde el punto de vista de la compatibilidad, sería razonable migrar las tablas de referencia de formato Excel a CSV. Los ficheros Excel pueden presentar problemas de compatibilidad en sistemas que no tengan instalado el soporte necesario, mientras que el formato CSV, al ser texto plano, funciona en cualquier entorno sin dependencias adicionales. Además, tener todas las tablas en CSV facilitaría tratarlas como una base de datos interna ligera, más sencilla de mantener y de versionar junto con el

resto del código. Esta migración sería asequible de implementar, ya que la función `load_all_tables()` de `help.py` centraliza toda la lógica de carga y bastaría con modificarla para leer ficheros CSV en lugar de Excel.

Por último, las tarifas y los datos normativos actualmente deben actualizarse de forma manual en las tablas de referencia. Conectar el modelo a fuentes de datos externas para que ciertos valores, como el precio de la electricidad o del gas, se actualicen de forma automática mantendría los resultados alineados con los precios vigentes sin intervención manual. Esto reduciría el riesgo de trabajar con datos desactualizados y aumentaría la utilidad de la herramienta como instrumento de análisis en tiempo real.

La pobreza energética es un problema que afecta a una parte significativa de los hogares españoles, y una de las dificultades para abordarlo es precisamente la de medirlo bien. Los indicadores estadísticos convencionales tienden a subestimar su magnitud real, y para diseñar intervenciones que funcionen hace falta poder estimar con precisión qué hogares no cubren sus necesidades energéticas y en qué medida. Ese es el problema que la metodología de la Cátedra de Energía y Pobreza lleva años tratando de resolver, y al que este proyecto ha intentado contribuir.

La migración del modelo a Python no es un fin en sí mismo, sino un medio para que esa metodología pueda aplicarse de forma más amplia, más sistemática y más accesible. Un modelo que puede ejecutarse de forma automática sobre bases de datos de miles de hogares, integrarse en plataformas digitales y ser utilizado por técnicos y trabajadores sociales sin conocimientos de programación tiene un alcance potencial muy distinto al de una herramienta que requiere software propietario y ejecución manual caso a caso. Ese salto en capacidad operativa es lo que este trabajo ha tratado de dar.

Bibliografía

[1] E. Arenas Pinilla, R. Barrella, J. I. Linares Hurtado y J. C. Romero Mora, «Investigación aplicada para el desarrollo de una herramienta web de cálculo del gasto eléctrico teórico para hogares españoles.,» Cátedra de Energía y Pobreza, Universidad Pontificia Comillas, inf. téc., 2021.

[2] E. Arenas Pinilla et al., «Caracterización del comportamiento energético en una muestra de hogares españoles,» ECODES; Cátedra de Energía y Pobreza, 2019.

[3] E. Arenas Pinilla, R. Barrella, Á. Cosín López-Medel, J. I. Linares Hurtado y J. C. Romero Mora, «Desarrollo de un modelo de cálculo de gasto eléctrico teórico en los hogares españoles,» ECODES; Cátedra de Energía y Pobreza, 2020.

[4] R. Barrella, J. I. Linares Hurtado, E. Arenas Pinilla y J. C. Romero Mora, «Estudio del gasto térmico teórico en refrigeración para hogares españoles,» ECODES – Fundación Ecología y Desarrollo / Cátedra de Energía y Pobreza, Universidad Pontificia Comillas, Madrid, inf. téc., nov. de 2022.

[5] ECODES. «¿Qué es la pobreza energética?» Consultado el 17 de mayo de 2026. dirección: https://ecodes.org/hacemos/energia-y-personas/cultura-energetica-y-pobreza-energetica/%20ni-un-hogar-sin-energia/que-es-la-pobreza-energetica

[6] Ministerio para la Transición Ecológica y el Reto Demográfico, «Estrategia Nacional contra la Pobreza Energética 2019-2024,» Gobierno de España, 2019. visitado 2025. dirección: https://www.miteco.gob.es/content/dam/miteco/es/ministerio/planes-estrategias/estrategia-pobreza-energetica/estrategianacionalcontralapobrezaenergetica_tcm30-502982.pdf

[7] J. C. Romero Mora, R. Barrella, E. Centeno Hernáez, S. Mora Rosado y A. Alguacil Denche, «Informe de Indicadores de Pobreza Energética en España 2024,» Cátedra de Energía y Pobreza, Universidad Pontificia Comillas, inf. téc. IIT-25-360I, nov. de 2025. dirección: https://www.iit.comillas.edu/documentacion/informetecnico/IIT-25-360I/Informe_de_Indicadores_de_pobreza_energ%C3%A9tica_en_Espa%C3%B1a_2024.pdf

[8] Instituto de Investigación Tecnológica, Universidad Pontificia Comillas, «Informe de Indicadores de Pobreza Energética en España 2024,» Universidad Pontificia Comillas, inf. téc., 2025. dirección: https://www.iit.comillas.edu/documentacion/informetecnico/IIT-25-360I/Informe_de_Indicadores_de_pobreza_energ%C3%A9tica_en_Espa%C3%B1a_2024.pdf

[9] Ministerio de Transportes, Movilidad y Agenda Urbana, «Documento Básico HE: Ahorro de Energía,» Código Técnico de la Edificación, inf. téc., 2022. dirección: https://www.codigotecnico.org/pdf/Documentos/HE/DccHE.pdf

Anexo II: Repositorio del proyecto

El código desarrollado durante este Trabajo Fin de Grado, junto con las tablas de referencia necesarias para ejecutar el modelo DIAGNÓSTICO en Python, se encuentra almacenado en un repositorio privado de GitHub. Se ha optado por mantener el repositorio en modo privado porque contiene la implementación completa del modelo y los ficheros de datos utilizados para reproducir los cálculos.

El acceso al repositorio puede solicitarse a través del correo universitario de la autora: `202202350@alu.comillas.edu`.

Para ejecutar la herramienta, basta con descargar la carpeta del proyecto, abrirla en un entorno de Python y lanzar la aplicación mediante el siguiente comando:

```
streamlit run app.py
```

El acceso al repositorio puede solicitarse a través del correo universitario de la autora: `202202350@alu.comillas.edu`.

Anexo III:Tablas de validación

		GENERAL			SISTEMAS TÉRMICOS				VIVIENDA			ELECTRODOMÉSTICOS										PERFIL OCUPANTE						
Caso	Provincia	Altitud	Tipo	Calif.	Fuente	Inst. calef.	Fuente	Inst. ACS	Hab.	Sup. (m²)	Área clim. (%)	Cocina	Horno	Micro.	Lavavaj.	Frigo.	Cong.	Lav.	Sec.	TV	PC	Móvil	Tablet	Ocup.	Par.	Estud.	Jub.	Incap.	Viudos	Tareas Hogar	Otros
1	Madrid	8000	Bloque	Ant. 1981	Gas Nat.	Indiv.	Gas Nat.	Indiv.	1	50	100	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	1	0	0	0	0	0	0	0
2	Barcelona	8000	Unif.	1981-2007	Gas Nat.	Central	Gas Nat.	Central	2	70	75	No	No	No	No	Sí	No	Sí	No	Sí	No	Sí	No	2	0	0	0	0	0	0	0
3	Sevilla	8000	Bloque	Post. 2007	Bomba calor	Indiv.	Elec.	Indiv.	3	80	50	NS /NC	NS /NC	NS /NC	NS /NC	NS /NC	NS /NC	NS /NC	NS /NC	NS /NC	NS /NC	NS /NC	NS /NC	0	1	1	0	0	0	1	0
4	Zaragoza	500	Unif.	A	Elec. (rad.)	Indiv.	Gas Nat.	Indiv.	4	100	100	Sí	No	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí	No	2	0	2	0	0	0	0	0
5	Bizkaia	8000	Bloque	B	Elec. (acum.	Indiv.	Elec.	Central	5	120	75	Sí	Sí	No	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	2	0	0	2	0	1	0	0
6	Valencia /València	8000	Unif.	C	Gasóleo	Indiv.	Gas Nat.	Indiv.	2	90	100	No	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	0	0	0	1	0	1	0	0
7	Málaga	8000	Bloque	D alta	glp	Indiv.	glp	Indiv.	3	70	50	Sí	No	No	Sí	Sí	No	Sí	No	Sí	Sí	Sí	No	1	1	1	0	0	0	0	0
8	Burgos	900	Unif.	D baja	Biomasa	Indiv.	Biomasa	Indiv.	4	150	100	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	NS /NC	NS /NC	Sí	NS /NC	2	0	0	0	1	0	1	0
9	Granada	700	Bloque	E alta	Carbón	Indiv.	Carbón	Indiv.	1	50	75	No	No	Sí	No	Sí	No	Sí	No	Sí	Sí	Sí	No	0	0	1	0	0	0	0	0
10	Coruña, A	8000	Unif.	E media	Gas Nat.	Central	Gas Nat.	Central	5	200	100	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	0	0	0	2	0	0	0	3
11	Alicante /Alacant	8000	Bloque	E baja	Bomba calor	Indiv.	Elec.	Indiv.	2	60	100	NS /NC	No	NS /NC	No	Sí	NS /NC	Sí	No	Sí	NS /NC	Sí	NS /NC	2	0	0	0	0	0	0	0
12	Huelva	8000	Unif.	F	Gas Nat.	Indiv.	Biomasa	Indiv.	3	80	50	Sí	Sí	No	Sí	Sí	No	Sí	Sí	Sí	No	Sí	Sí	1	2	0	0	0	0	0	0
13	Ceuta	8000	Bloque	G	glp	Indiv.	glp	Indiv.	4	90	75	Sí	No	Sí	No	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	2	0	2	0	0	0	0	0
14	Melilla	8000	Unif.	Ant. 1981	Gasóleo	Indiv.	Elec.	Indiv.	2	110	100	No	NS /NC	No	NS /NC	Sí	No	Sí	NS /NC	Sí	No	Sí	No	0	0	0	0	0	0	2	0
15	Salamanca	800	Bloque	1981-2007	Elec. (rad.)	Ap.	Gas Nat.	Indiv.	3	75	100	Sí	Sí	Sí	Sí	Sí	No	Sí	No	Sí	Sí	Sí	Sí	1	1	1	0	0	0	0	0
16	Lleida	1000	Unif.	C	Gas Nat.	Indiv.	Carbón	Indiv.	5	180	75	Sí	Sí	No	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	3	0	2	0	0	0	0	0
17	Murcia	8000	Bloque	D alta	Bomba calor	Indiv.	Gas Nat.	Central	4	85	50	No	No	No	No	Sí	No	Sí	No	Sí	No	Sí	No	2	0	0	2	0	0	0	0
18	Huesca	1200	Unif.	E alta	Biomasa	Central	Gas Nat.	Indiv.	2	130	100	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	2	0	0	0	0	0	0	0
19	Almería	8000	Bloque	F	Elec. (acum.	Ap.	Elec.	Indiv.	1	45	100	NS /NC	No	NS /NC	No	Sí	No	Sí	No	Sí	NS /NC	Sí	No	0	0	0	1	0	0	0	0
20	Pontevedra	400	Unif.	G	Gas Nat.	Central	glp	Indiv.	3	95	75	Sí	No	Sí	Sí	Sí	No	Sí	No	NS /NC	Sí	Sí	NS /NC	1	0	1	1	0	0	0	0
21	Toledo	600	Bloque	B	Carbón	Indiv.	Gas Nat.	Indiv.	5	100	100	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	Sí	2	1	0	0	1	0	1	0
22	Cádiz	8000	Inifamili	A	Elec. (rad.)	Indiv.	Elec.	Indiv.	4	160	50	No	Sí	No	No	Sí	NS /NC	Sí	NS /NC	Sí	No	Sí	Sí	0	0	0	0	0	2	2	0
23	Teruel	1100	Bloque	1981-2007	Gasóleo	Central	Biomasa	Indiv.	2	65	100	Sí	No	Sí	No	Sí	No	Sí	No	Sí	Sí	Sí	No	0	0	0	2	0	0	0	0
24	Córdoba	8000	Unif.	D baja	Bomba calor	Ap.	Gas Nat.	Indiv.	3	120	75	Sí	Sí	Sí	Sí	Sí	Sí	Sí	No	Sí	Sí	Sí	Sí	2	1	0	0	0	0	0	0
25	Asturias	300	Bloque	E media	Gas Nat.	Central	Elec.	Central	4	85	100	NS /NC	NS /NC	NS /NC	NS /NC	Sí	NS /NC	Sí	NS /NC	NS /NC	NS /NC	Sí	NS /NC	0	2	2	0	0	0	0	0

Figura 1: Casos de validación: datos de entrada completos.

Ant. = Anterior a 1981; Post. = Posterior a 2007; Gas Nat. = Gas Natural; Elec. (rad.) = Electricidad (radiadores); Elec. (acum.) = Electricidad (acumuladores); Indiv. = Individual; Ap. = Aparatos; Hab. = Habitantes; Sup. = Superficie; Área clim. = Área climatizada; Cong. = Congelador; Lav. = Lavadora; Sec. = Secadora; Ocup. = Ocupados; Par. = Parados; Estud. = Estudiantes; Jub. = Jubilados; Incap. = Incapacitados.

		PYTHON: DEMANDA (kWh)				PYTHON: CONSUMO (kWh)				PYTHON: GASTO (€)								EES: DEMANDA (kWh)				EES: CONSUMO (kWh)				EES: GASTO (€)							
Caso	Provincia	Dem. Calef.	Dem. Refrig.	Dem. ACS	Dem. Elec. noterm.	Cons. Calef.	Cons. ACS	Cons. Refrig.	Cons. Elec. noterm.	Gasto CAL	Gasto ACS	Gasto Refrig.	Gasto Combust.	Elec. noterm sinREF	Elec. noterm conREF	Elec. total sinREF	Elec. total conREF	Dem. Calef.	Dem. Refrig.	Dem. ACS	Dem. Elec. noterm.	Cons. Calef.	Cons. ACS	Cons. Elec. noterm.	Cons. Refrig.	Gasto CAL	Gasto ACS	Gasto Refrig.	Gasto Combust.	Elec. noterm sinREF	Elec. noterm conREF	Elec. total sinREF	Elec. total conREF
1	Madrid	7.953	1.759	409	2.768	10.604	532	504	2.768	724	94	132,05	817,29	792,93	964,56	792,93	1.096,61	7953	1759	408,6	2768	10604	532,1	2768	504,1	723,52	93,72	132,05	817,24	792,87	964,48	792,87	1096,53
2	Barcelona	6.318	814	1.097	1.624	9.025	1.355	221	1.624	470	61	58,48	530,28	542,13	679,44	542,13	737,91	6318	814,2	1097	1624	9025	1355	1624	220,6	469,77	60,5	58,48	530,27	542,12	679,41	542,12	737,89
3	Sevilla	1.176	592	1.161	3.666	302	1.172	156	3.666	67	284	40,10	0,00	1.188,45	1.188,45	1.539,37	1.579,47	1176	592,5	1161	3666	301,6	1172	3666	155,5	66,54	284,38	40,1	0	1188,43	1188,43	1539,35	1579,45
4	Zaragoza	2.860	899	2.288	3.807	2.860	2.706	229	3.807	646	289	60,08	288,74	1.257,74	1.257,74	1.903,30	1.963,38	2860	899	2288	3806	2860	2706	3806	229,3	645,55	288,74	60,08	288,74	1257,65	1257,65	1903,21	1963,29
5	Bizkaia	1.528	0	2.054	5.132	1.528	2.075	0	5.132	293	503	0,00	0,00	1.484,12	1.484,12	2.280,00	2.280,00	1528	0	2054	5132	1528	2075	5132	0	292,63	503,25	0	0	1484,02	1484,02	2279,91	2279,91
6	Valencia /València	2.224	1.562	1.077	3.136	2.679	1.274	399	3.136	150	204	104,42	353,76	919,07	1.056,37	919,07	1.160,79	2224	1562	1077	3136	2679	1274	3136	398,6	149,76	204,01	104,42	353,77	919	1058,29	919	1160,71
7	Málaga	1.036	552	1.161	3.341	1.381	1.511	158	3.341	112	123	41,44	235,21	989,10	1.107,33	989,10	1.148,77	1036	552,1	1161	3341	1381	1511	3341	158,2	112,28	122,89	41,44	235,17	989,11	1107,33	989,11	1148,77
8	Burgos	19.595	0	2.367	4.942	55.984	6.763	0	4.942	2.609	315	0,00	2.924,03	1.407,15	1.407,15	1.407,15	1.407,15	19590	0	2367	4942	55972	6763	4942	0	2608,3	315,16	0	2923,44	1407,57	1407,57	1407,57	1407,57
9	Granada	3.083	863	407	1.264	7.708	1.018	247	1.264	237	31	64,79	268,77	418,09	589,72	418,09	654,50	3083	863,1	407,2	1264	7708	1018	1264	247,3	237,42	31,35	64,79	268,77	418,08	589,68	418,08	654,47
10	Coruña, A	20.204	0	2.796	5.559	28.863	3.455	0	5.559	1.356	154	0,00	1.509,77	1.583,73	1.583,73	1.583,73	1.583,73	20204	0	2796	5558	28863	3455	5558	0	1355,5	154,26	0	1509,77	1583,66	1583,66	1583,66	1583,66
11	Alicante /Alacant	3.536	2.145	777	2.328	1.105	784	633	2.328	244	190	163,21	0,00	820,59	854,91	1.254,67	1.452,20	3536	2145	776,6	2328	1105	784,5	2328	632,8	243,79	190,29	163,21	0	820,53	854,85	1254,62	1452,15
12	Huelva	5.714	2.272	1.580	3.513	7.619	4.513	670	3.513	572	210	172,86	782,81	1.031,92	1.150,16	1.031,92	1.323,02	5714	2272	1580	3513	7619	4513	3513	670,3	572,45	210,33	172,86	782,78	1031,93	1150,15	1031,93	1323,02
13	Ceuta	6.576	2.599	1.594	3.638	8.768	2.076	745	3.638	714	169	162,84	883,23	903,33	986,10	903,33	1.148,94	6576	2599	1594	3638	8768	2075	3638	744,7	714,14	169,04	162,84	883,18	903,26	986,03	903,26	1148,86
14	Melilla	9.117	5.388	1.049	2.338	13.024	1.060	1.544	2.338	608	215	337,58	607,68	601,18	715,79	815,76	1.267,95	9117	5388	1049	2338	13024	1060	2338	1544	607,71	214,58	337,59	607,71	601,17	715,76	815,75	1267,93
15	Salamanca	7.445	800	1.276	3.543	7.837	1.661	217	3.543	1.769	227	57,50	226,93	1.157,67	1.157,67	2.926,57	2.984,06	7445	800,5	1276	3543	7837	1661	3543	216,9	1768,9	226,92	57,5	226,92	1157,67	1157,67	2926,57	2984,07
16	Lleida	14.875	0	3.021	5.429	16.346	4.231	0	5.429	1.014	130	0,00	1.144,52	1.551,53	1.551,53	1.551,53	1.551,53	14875	0	3021	5429	16346	4231	5429	0	1014,1	130,32	0	1144,46	1551,46	1551,46	1551,46	1551,46
17	Murcia	1.905	670	1.576	2.279	595	1.948	192	2.279	131	87	50,32	86,98	877,11	877,11	1.008,46	1.058,79	1905	670,4	1576	2279	595,2	1948	2279	192,1	131,3	86,98	50,32	86,98	877,08	877,08	1008,38	1058,7
18	Huesca	20.374	0	1.212	3.633	50.934	1.578	0	3.633	2.374	222	0,00	2.595,54	1.042,94	1.042,94	1.042,94	1.042,94	20374	0	1212	3633	50934	1578	3633	0	2373,5	221,99	0	2595,53	1042,88	1042,88	1042,88	1042,88
19	Almería	2.191	1.852	385	1.719	2.306	389	546	1.719	442	94	140,86	0,00	534,87	703,20	1.070,87	1.380,06	2191	1852	385,4	1719	2306	389,3	1719	546,2	441,57	94,44	140,86	0	534,85	703,16	1070,86	1380,03
20	Pontevedra	21.178	0	1.722	3.494	30.254	2.243	0	3.494	1.418	182	0,00	1.599,98	1.027,36	1.027,36	1.027,36	1.027,36	21178	0	1722	3494	30254	2243	3494	0	1417,6	182,37	0	1599,98	1027,37	1027,37	1027,37	1027,37
21	Toledo	1.917	720	2.010	5.624	2.739	2.378	184	5.624	84	269	48,13	353,71	1.600,07	1.676,35	1.600,07	1.724,48	1917	720,2	2010	5624	2739	2378	5624	183,7	84,36	269,35	48,13	353,71	1600	1676,27	1600	1724,4
22	Cádiz	822	719	2.110	3.152	822	2.132	183	3.152	180	517	48,06	0,00	1.094,70	1.094,70	1.791,37	1.839,44	821,8	719,2	2110	3152	821,8	2132	3152	183,5	179,63	517,05	48,06	0	1094,66	1094,66	1791,33	1839,4
23	Teruel	8.554	0	865	2.609	13.159	2.472	0	2.609	736	115	0,00	850,82	787,65	787,65	787,65	787,65	8554	0	865,3	2609	13159	2472	2609	0	735,64	115,21	0	850,84	787,62	787,62	787,62	787,62
24	Córdoba	5.416	3.448	1.597	3.984	1.692	2.080	1.017	3.984	373	252	262,34	251,71	1.267,66	1.267,66	1.641,04	1.903,38	5414	3448	1597	3984	1692	2080	3984	1017	373,28	251,7	262,34	251,7	1267,63	1267,63	1640,91	1903,25
25	Asturias	9.584	0	1.687	4.036	13.691	1.704	0	4.036	678	413	0,00	678,10	1.181,33	1.181,33	1.594,61	1.594,61	9584	0	1687	4036	13691	1704	4036	0	678,1	413,28	0	678,1	1181,29	1181,29	1594,57	1594,57

Figura 2: Resultados de demanda, consumo y gasto obtenidos por Python y EES para los 25 casos de validación.

CASO		ERROR (PY - EES)							
Caso	Provincia	Δ Demanda Calefacción	% Error Demanda Calefacción	Δ Demanda Refrigeración	% Error Demanda Refrigeración	Δ Demanda ACS	% Error Demanda ACS	Δ Demanda Electricidad No térmica	% Error Demanda Electricidad No térmica
1	Madrid	0,12	0,0015%	0,19	0,0108%	0,01	0,0024%	0,07	0,0025%
2	Barcelona	0,17	0,0027%	0,04	0,0049%	0,39	0,0356%	0,17	0,0105%
3	Sevilla	0,30	0,0255%	0,02	0,0034%	0,38	0,0327%	0,37	0,0101%
4	Zaragoza	0,00	0,0000%	0,00	0,0000%	0,42	0,0184%	0,67	0,0176%
5	Bizkaia	0,09	0,0059%	0,00	0,0000%	0,09	0,0044%	0,43	0,0084%
6	Valencia /València	0,44	0,0198%	0,40	0,0256%	0,15	0,0139%	0,35	0,0112%
7	Málaga	0,01	0,0010%	0,01	0,0018%	0,38	0,0327%	0,17	0,0051%
8	Burgos	4,53	0,0231%	0,00	0,0000%	0,09	0,0038%	0,16	0,0032%
9	Granada	0,31	0,0101%	0,02	0,0023%	0,03	0,0074%	0,19	0,0150%
10	Coruña, A	0,25	0,0012%	0,00	0,0000%	0,15	0,0054%	0,59	0,0106%
11	Alicante /Alacant	0,28	0,0079%	0,36	0,0168%	0,05	0,0064%	0,04	0,0017%
12	Huelva	0,42	0,0074%	0,33	0,0145%	0,29	0,0184%	0,31	0,0088%
13	Ceuta	0,20	0,0030%	0,14	0,0054%	0,13	0,0082%	0,05	0,0014%
14	Melilla	0,46	0,0050%	0,20	0,0037%	0,19	0,0181%	0,45	0,0192%
15	Salamanca	0,06	0,0008%	0,02	0,0025%	0,18	0,0141%	0,16	0,0045%
16	Lleida	0,20	0,0013%	0,00	0,0000%	0,05	0,0017%	0,36	0,0066%
17	Murcia	0,24	0,0126%	0,00	0,0000%	0,48	0,0305%	0,31	0,0136%
18	Huesca	0,29	0,0014%	0,00	0,0000%	0,22	0,0182%	0,44	0,0121%
19	Almería	0,48	0,0219%	0,43	0,0232%	0,03	0,0078%	0,20	0,0116%
20	Pontevedra	0,22	0,0010%	0,00	0,0000%	0,43	0,0250%	0,36	0,0103%
21	Toledo	0,35	0,0183%	0,03	0,0042%	0,44	0,0219%	0,14	0,0025%
22	Cádiz	0,04	0,0049%	0,00	0,0000%	0,23	0,0109%	0,42	0,0133%
23	Teruel	0,37	0,0043%	0,00	0,0000%	0,02	0,0023%	0,02	0,0008%
24	Córdoba	1,94	0,0358%	0,49	0,0142%	0,49	0,0307%	0,20	0,0050%
25	Asturias	0,01	0,0001%	0,00	0,0000%	0,29	0,0172%	0,29	0,0072%
			0,0087%		0,0053%		0,0155%		0,0085%
		Max	0,0358%	Max	0,0256%	Max	0,0356%	Max	0,0192%
		MAE demanda				0,0095%			

Figura 3: Errores relativos porcentuales y MAE por variable y caso para demanda.

CASO		ERROR (PY - EES)							
Caso	Provincia	Δ Consumo Calefacción	% Error Consumo Calefacción	Δ Consumo ACS	% Error Consumo ACS	Δ Consumo Refrigeración	% Error Consumo Refrigeración	Δ Consumo Electricidad no térmica	% Error Consumo Electricidad no térmica
1	Madrid	0,16	0,0015%	0,01	0,0019%	0,03	0,0227%	0,07	0,0025%
2	Barcelona	0,47	0,0052%	0,17	0,0125%	0,04	0,0684%	0,17	0,0105%
3	Sevilla	0,02	0,0066%	0,35	0,0299%	0,01	0,0249%	0,37	0,0101%
4	Zaragoza	0,00	0,0000%	0,23	0,0085%	0,04	0,0666%	0,67	0,0176%
5	Bizkaia	0,09	0,0059%	0,34	0,0164%	0,00	0,0000%	0,43	0,0084%
6	Valencia /València	0,01	0,0004%	0,08	0,0063%	0,03	0,0287%	0,35	0,0112%
7	Málaga	0,34	0,0246%	0,43	0,0285%	0,01	0,0241%	0,17	0,0051%
8	Burgos	12,38	0,0221%	0,11	0,0016%	0,00	0,0000%	0,16	0,0032%
9	Granada	0,28	0,0036%	0,08	0,0079%	0,00	0,0000%	0,19	0,0150%
10	Coruña, A	0,22	0,0008%	0,08	0,0023%	0,00	0,0000%	0,59	0,0106%
11	Alicante /Alacant	0,09	0,0081%	0,01	0,0013%	0,05	0,0306%	0,04	0,0017%
12	Huelva	0,23	0,0030%	0,44	0,0097%	0,00	0,0000%	0,31	0,0088%
13	Ceuta	0,26	0,0030%	0,62	0,0299%	0,04	0,0246%	0,05	0,0014%
14	Melilla	0,37	0,0028%	0,22	0,0208%	0,22	0,0652%	0,45	0,0192%
15	Salamanca	0,22	0,0028%	0,44	0,0265%	0,03	0,0522%	0,16	0,0045%
16	Lleida	0,37	0,0023%	0,16	0,0038%	0,00	0,0000%	0,36	0,0066%
17	Murcia	0,19	0,0319%	0,19	0,0098%	0,01	0,0199%	0,31	0,0136%
18	Huesca	0,28	0,0005%	0,05	0,0032%	0,00	0,0000%	0,44	0,0121%
19	Almería	0,19	0,0082%	0,02	0,0051%	0,01	0,0071%	0,2	0,0116%
20	Pontevedra	0,03	0,0001%	0,04	0,0018%	0,00	0,0000%	0,36	0,0103%
21	Toledo	0,08	0,0029%	0,38	0,0160%	0,02	0,0416%	0,14	0,0025%
22	Cádiz	0,04	0,0049%	0,46	0,0216%	0,03	0,0624%	0,42	0,0133%
23	Teruel	0,44	0,0033%	0,22	0,0089%	0,00	0,0000%	0,02	0,0008%
24	Córdoba	0,48	0,0284%	0,33	0,0159%	0,25	0,0953%	0,2	0,0050%
25	Asturias	0,41	0,0030%	0,25	0,0147%	0,00	0,0000%	0,29	0,0072%
			0,0070%		0,0122%		0,0254%		0,0085%
		Max	0,0319%	Max	0,0299%	Max	0,0953%	Max	0,0192%
		MAE consumo				0,0133%			

Figura 4: Errores relativos porcentuales y MAE por variable y caso para consumo.

CASO		ERROR (PY - EES)															
Caso	Provincia	Δ Gasto Calefacción	% Error Gasto Calefacción	Δ Gasto ACS	% Error Gasto ACS	Δ Gasto Refrigeración	% Error Gasto Refrigeración	Δ Gasto Combustibles	% Error Gasto Combustibles	Δ Electricidad no térmica sinREF	% Error Electricidad no térmica sinREF	Δ Electricidad no térmica conREF	% Error Electricidad no térmica conREF	Δ Elecricidad total sinREF	% Error Electricidad total sinREF	Δ Electricidad total conREF	% Error Electricidad total conREF
1	Madrid	0,05	0,0069%	0,00	0,0000%	0,00	0,0000%	0,05	0,0061%	0,06	0,0076%	0,08	0,0083%	0,06	0,0076%	0,08	0,0073%
2	Barcelona	0,00	0,0000%	0,01	0,0165%	0,00	0,0000%	0,01	0,0019%	0,01	0,0018%	0,03	0,0044%	0,01	0,0018%	0,02	0,0027%
3	Sevilla	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,02	0,0017%	0,02	0,0017%	0,02	0,0013%	0,02	0,0013%
4	Zaragoza	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,09	0,0072%	0,09	0,0072%	0,09	0,0047%	0,09	0,0046%
5	Bizkaia	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,10	0,0067%	0,10	0,0067%	0,09	0,0039%	0,09	0,0039%
6	Valencia /València	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,01	0,0028%	0,07	0,0076%	0,08	0,0076%	0,07	0,0076%	0,08	0,0069%
7	Málaga	0,04	0,0356%	0,00	0,0000%	0,00	0,0000%	0,04	0,0170%	0,01	0,0010%	0,00	0,0000%	0,01	0,0010%	0,00	0,0000%
8	Burgos	0,59	0,0226%	0,00	0,0000%	0,00	0,0000%	0,59	0,0202%	0,42	0,0298%	0,42	0,0298%	0,42	0,0298%	0,42	0,0298%
9	Granada	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,01	0,0024%	0,04	0,0068%	0,01	0,0024%	0,03	0,0046%
10	Coruña, A	0,00	0,0000%	0,01	0,0065%	0,00	0,0000%	0,00	0,0000%	0,07	0,0044%	0,07	0,0044%	0,07	0,0044%	0,07	0,0044%
11	Alicante /Alacant	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,06	0,0073%	0,06	0,0070%	0,05	0,0040%	0,05	0,0034%
12	Huelva	0,04	0,0070%	0,00	0,0000%	0,00	0,0000%	0,03	0,0038%	0,01	0,0010%	0,01	0,0009%	0,01	0,0010%	0,00	0,0000%
13	Ceuta	0,04	0,0056%	0,02	0,0118%	0,00	0,0000%	0,05	0,0057%	0,07	0,0077%	0,07	0,0071%	0,07	0,0077%	0,08	0,0070%
14	Melilla	0,03	0,0049%	0,00	0,0000%	0,01	0,0030%	0,03	0,0049%	0,01	0,0017%	0,03	0,0042%	0,01	0,0012%	0,02	0,0016%
15	Salamanca	0,00	0,0000%	0,01	0,0044%	0,00	0,0000%	0,01	0,0044%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,01	0,0003%
16	Lleida	0,06	0,0059%	0,00	0,0000%	0,00	0,0000%	0,06	0,0052%	0,07	0,0045%	0,07	0,0045%	0,07	0,0045%	0,07	0,0045%
17	Murcia	0,05	0,0381%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,03	0,0034%	0,03	0,0034%	0,08	0,0079%	0,09	0,0085%
18	Huesca	0,00	0,0000%	0,01	0,0045%	0,00	0,0000%	0,01	0,0004%	0,06	0,0058%	0,06	0,0058%	0,06	0,0058%	0,06	0,0058%
19	Almería	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,02	0,0037%	0,04	0,0057%	0,01	0,0009%	0,03	0,0022%
20	Pontevedra	0,00	0,0000%	0,01	0,0055%	0,00	0,0000%	0,00	0,0000%	0,01	0,0010%	0,01	0,0010%	0,01	0,0010%	0,01	0,0010%
21	Toledo	0,00	0,0000%	0,01	0,0037%	0,00	0,0000%	0,00	0,0000%	0,07	0,0044%	0,08	0,0048%	0,07	0,0044%	0,08	0,0046%
22	Cádiz	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,04	0,0037%	0,04	0,0037%	0,04	0,0022%	0,04	0,0022%
23	Teruel	0,03	0,0041%	0,00	0,0000%	0,00	0,0000%	0,02	0,0024%	0,03	0,0038%	0,03	0,0038%	0,03	0,0038%	0,03	0,0038%
24	Córdoba	0,10	0,0268%	0,01	0,0040%	0,00	0,0000%	0,01	0,0040%	0,03	0,0024%	0,03	0,0024%	0,13	0,0079%	0,13	0,0068%
25	Asturias	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,00	0,0000%	0,04	0,0034%	0,04	0,0034%	0,04	0,0025%	0,04	0,0025%
			0,0063%		0,0023%		0,0001%		0,0032%		0,0050%		0,0054%		0,0048%		0,0048%
		Max	0,0381%	Max	0,0165%	Max	0,0030%	Max	0,0202%	Max	0,0298%	Max	0,0298%	Max	0,0298%	Max	0,0298%
		MAE gastos											0,0040%				

Figura 5: Errores relativos porcentuales y MAE por variable y caso para gasto.

Caso	Provincia	ZONA 1	ZONA 2	ZONA 3	ZONA 4
1	Madrid				
2	Barcelona		0,0684%		
3	Sevilla				0,0249%
4	Zaragoza			0,0666%	
5	Bizkaia	0,0000%			
6	Valencia /València			0,0287%	
7	Málaga			0,0241%	
8	Burgos	0,0000%			
9	Granada			0,0000%	
10	Coruña, A	0,0000%			
11	Alicante /Alacant				0,0306%
12	Huelva				0,0000%
13	Ceuta			0,0246%	
14	Melilla			0,0652%	
15	Salamanca		0,0522%		
16	Lleida	0,0000%			
17	Murcia			0,0199%	
18	Huesca	0,0000%			
19	Almería				0,0071%
20	Pontevedra	0,0000%			
21	Toledo			0,0416%	
22	Cádiz			0,0624%	
23	Teruel				0,0000%
24	Córdoba				0,0953%
25	Asturias	0,0000%			
	Media error	0,0000%	0,0603%	0,0370%	0,0226%

Figura 6: Errores relativos porcentuales por zona climática de verano.

Anexo IV: Alineación con los Objetivos de Desarrollo Sostenible

Este trabajo se enmarca en el ámbito del diagnóstico de la pobreza energética en hogares españoles y contribuye, de forma directa o indirecta, a varios Objetivos de Desarrollo Sostenible (ODS) de la Agenda 2030 de Naciones Unidas. A continuación se describe la relación del proyecto con cada uno de ellos.

.1. *ODS 1: FIN DE LA POBREZA*

La pobreza energética es una forma concreta de pobreza: afecta a hogares que no pueden cubrir sus necesidades básicas de energía debido a sus ingresos insuficientes, a la ineficiencia de su vivienda, o a una combinación de ambos factores. No es un fenómeno marginal; según los indicadores de la Estrategia Nacional contra la Pobreza Energética, una parte significativa de los hogares españoles destina una proporción desproporcionada de sus ingresos a cubrir el gasto energético, o directamente no llega a cubrirlo.

El modelo desarrollado en este trabajo permite estimar el gasto energético teórico de cualquier hogar a partir de sus características reales: ubicación, superficie, antigüedad del edificio, instalaciones y composición del hogar. Esto facilita identificar qué hogares se encuentran en situación de vulnerabilidad y en qué medida, algo que no es posible con indicadores agregados. Al migrar el modelo a Python, se amplía su capacidad para aplicarse de forma sistemática a grandes bases de datos, algo que con EES no era viable por las limitaciones de integración y escalabilidad de esa herramienta. En la práctica, esto significa que organizaciones sociales o administraciones públicas pueden utilizar el modelo para priorizar intervenciones y orientar mejor las políticas de apoyo a los hogares más vulnerables.

.2. *ODS 7: ENERGÍA ASEQUIBLE Y NO CONTAMINANTE*

Uno de los pilares del ODS 7 es garantizar el acceso universal a servicios energéticos asequibles. Para saber si ese acceso existe en la práctica, hace falta poder estimar cuánto debería gastar un hogar para cubrir sus necesidades energéticas en condiciones de confort, y compararlo con lo que realmente gasta o puede gastar. Eso es exactamente lo que calcula el modelo: un gasto teórico de referencia que sirve como punto de comparación.

Cuando este modelo se aplicó a la base de datos del programa «Ni un hogar sin energía», el consumo real de los hogares vulnerables resultó entre un 18 % y un 20 % inferior al teórico, lo que sugiere que una parte de esos hogares no cubría sus necesidades mínimas. Disponer de una herramienta más flexible y accesible facilita que este tipo de análisis pueda realizarse con mayor frecuencia y sobre muestras más amplias, contribuyendo a una mejor comprensión de las barreras de acceso a la energía en España.

.3. ODS 10: REDUCCIÓN DE LAS DESIGUALDADES

La pobreza energética no afecta por igual a todos los hogares. Incide de forma más intensa en los de menor renta, en viviendas antiguas con peor aislamiento y en zonas climáticas con mayor demanda térmica, ya sea por frío en invierno o por calor en verano. El modelo permite cruzar todas estas variables y cuantificar en qué medida cada factor contribuye al gasto energético, lo que resulta útil para diseñar intervenciones más precisas: no es lo mismo priorizar la el aislamiento en una zona E1 que subvencionar el cambio de caldera en una zona A3.

La migración a Python facilita además que el modelo pueda integrarse en plataformas web accesibles a profesionales del ámbito social, como trabajadores sociales o técnicos municipales, que no necesariamente disponen de herramientas de ingeniería. Ampliar el alcance de este tipo de análisis más allá del entorno académico es, en sí mismo, una forma de reducir la desigualdad en el acceso a información relevante para la toma de decisiones.

.4. ODS 11: CIUDADES Y COMUNIDADES SOSTENIBLES

El ODS 11 promueve entornos urbanos más eficientes y sostenibles, lo que pasa en buena medida por mejorar el comportamiento energético del parque residencial existente. El modelo desarrollado permite analizar ese comportamiento a escala de hogar individual y, si se aplica sobre una muestra representativa, agregarlo para obtener una imagen del parque residencial a nivel de ciudad o comunidad autónoma.

Esto puede ser útil, por ejemplo, para identificar en qué zonas se concentran los edificios con peor aislamiento, qué sistemas de calefacción o ACS predominan en cada zona climática, o qué medidas de rehabilitación tendrían mayor impacto sobre el gasto energético de los hogares más vulnerables. En ese sentido, la herramienta puede ser un apoyo concreto para la planificación urbana orientada a la eficiencia energética, especialmente en municipios que estén desarrollando estrategias locales de energía o planes de acción climática.