



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

ANÁLISIS ECONÓMICO DEL SUMINISTRO ENERGÉTICO EN CENTROS DE DATOS DEDICADOS A INTELIGENCIA ARTIFICIAL

Autor: Carlota Otero Galeano

Director: Javier Ponce Cintas

Madrid

Julio de 2026

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título

Análisis Económico del Suministro Energético en Centros de Datos dedicados a
Inteligencia Artificial

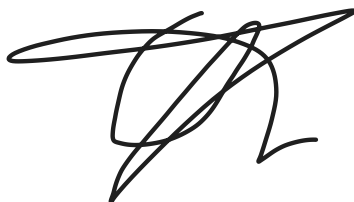
en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2025/2026 es de mi autoría, original e inédito 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.



Fdo.: Carlota Otero Galeano

Fecha: 19/ 07/ 2026

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Javier Ponce Cintas

Fecha: 19/ 07/ 2026



GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

TRABAJO FIN DE GRADO

ANÁLISIS ECONÓMICO DEL SUMINISTRO ENERGÉTICO EN CENTROS DE DATOS DEDICADOS A INTELIGENCIA ARTIFICIAL

Autor: Carlota Otero Galeano

Director: Javier Ponce Cintas

Madrid

Julio de 2026

ANÁLISIS ECONÓMICO DEL SUMINISTRO ENERGÉTICO EN CENTROS DE DATOS DEDICADOS A INTELIGENCIA ARTIFICIAL

Autor: Carlota Otero Galeano

Director: Javier Ponce Cintas

Entidad Colaboradora: ICAI – Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

Este Trabajo Fin de Grado estudia la optimización del suministro eléctrico de un centro de datos de 20 MW de potencia IT dedicado a inteligencia artificial, conectado al mercado mayorista español (OMIE) y evaluado con datos horarios del año 2025. Para ello se plantean siete modelos de optimización en Python, formulados con Pyomo y resueltos con HiGHS, que incorporan de forma incremental distintos grados de libertad: la posibilidad de mover en el tiempo la carga de entrenamiento, un sistema de refrigeración modelado a partir de la temperatura exterior, una batería de gran capacidad que aprovecha la volatilidad horaria del mercado y un sistema de almacenamiento térmico de frío (CTES) que realiza ese mismo arbitraje sobre la demanda de refrigeración y, por último, un sistema híbrido que combina ambos almacenamientos. Cada modelo se evalúa mediante indicadores económicos como el VAN, la TIR, el periodo de retorno y un análisis de sensibilidad sobre los parámetros más inciertos.

Palabras clave: Centros de datos, Inteligencia artificial, Optimización energética, Mercado eléctrico OMIE, Flexibilidad de demanda, Almacenamiento en batería, CTES, Refrigeración por amoníaco.

1. Introducción

La inteligencia artificial ha pasado en pocos años a ser una pieza central de la economía digital, y los centros de datos que la sostienen funcionan prácticamente 24/7 y a plena potencia. Los últimos informes sitúan ya su consumo en torno al 1,5 % de la electricidad mundial, y todo apunta a que la cifra va a seguir creciendo a medida que se generalicen

los modelos de gran tamaño. Para el operador, la consecuencia es que la factura eléctrica se ha convertido en una de las partidas más pesadas de su cuenta de explotación.

El mercado eléctrico mayorista español, gestionado por OMIE, tiene precios horarios muy variables, y esa variabilidad puede jugar tanto a favor como en contra del operador. Si una parte de la carga puede esperar, como ocurre con ciertos trabajos de entrenamiento, o si se dispone de almacenamiento, es posible desplazar consumo hacia horas baratas y reducir la compra en horas caras. De ahí surge la pregunta principal del trabajo: cuánto se puede ahorrar realmente con este tipo de estrategias y a partir de qué punto compensa la inversión que necesitan.

2. Definición del proyecto

El objetivo del trabajo es construir varios modelos de optimización del suministro eléctrico de un centro de datos de IA y, a partir de ellos, medir el ahorro y la rentabilidad de cada estrategia. El proceso se organiza en tres bloques. El primero prepara el caso base, con los precios horarios de OMIE y el perfil de demanda durante las 8.760 horas del año. El segundo formula siete modelos de optimización, cada uno con un grado de libertad adicional respecto al anterior. Los Modelos 1 a 4 se plantean como problemas lineales continuos, mientras que los Modelos 5 a 7 incorporan variables binarias para evitar la carga y descarga simultáneas de los sistemas de almacenamiento, por lo que se resuelven como problemas lineales enteros mixtos (MILP). El tercer bloque evalúa económicamente cada caso mediante indicadores de rentabilidad y análisis de sensibilidad.

3. Descripción del modelo/sistema/herramienta

El centro modelado tiene 20 MW de potencia IT y un PUE de 1,3, lo que da una potencia total de 26 MW. Su carga se reparte entre dos tipos de tarea: el entrenamiento, flexible y responsable del 60 % del consumo IT, y la inferencia, rígida y responsable del 40 % restante. Sobre ese caso base se han desarrollado siete modelos:

— Modelo 1: reproduce la operación sin gestión y fija el coste de referencia.

- Modelo 2: permite mover el entrenamiento a lo largo de todo el año, una cota teórica no realista en la práctica.
- Modelo 3: limita esa flexibilidad al día, que sí es operativamente viable.
- Modelo 4: sustituye la climatización del dataset por un modelo calibrado con la temperatura exterior.
- Modelo 5: añade una batería BESS de 20 MWh que hace arbitraje horario en el mercado eléctrico.
- Modelo 6: sustituye esa batería por un sistema de almacenamiento térmico de frío (CTES), que desplaza la demanda de refrigeración a las horas más baratas con un CAPEX menor y sin necesidad de reemplazo en el horizonte de análisis.
- Modelo 7: combina la batería BESS y el tanque CTES sobre la misma base, coordinando ambos almacenamientos junto con la flexibilidad del training; alcanza el mayor ahorro absoluto de todos los modelos, aunque representa una rentabilidad por euro invertido inferior a la del CTES en solitario.

Todos los modelos se han implementado en Python con Pyomo y se han resuelto con el solver HiGHS.

4. Resultados

El coste anual del caso base asciende a 8.903.875 €. El Modelo 2, con flexibilidad anual del entrenamiento, reduce el coste hasta 7.358.840 €, lo que supone un ahorro de 1.545.035 € (17,35 %). Este resultado representa una cota teórica, ya que permite desplazar el entrenamiento libremente a lo largo de todo el año. Al limitar esa flexibilidad al interior de cada día, el Modelo 3 obtiene un coste de 8.080.644 €, con un ahorro de 823.231 € (9,25 %) sin necesidad de inversión de capital.

El Modelo 4, que incorpora la refrigeración dependiente de la temperatura exterior, obtiene un coste anual de 8.081.904 €, prácticamente igual al del Modelo 3. Esto indica que el modelo climático reproduce de forma coherente el perfil original de climatización,

pero no genera por sí solo un ahorro adicional relevante. El Modelo 5, con batería BESS, reduce el coste anual hasta 7.464.296 €, con un ahorro de 1.439.579 € (16,17 %). Sin embargo, su CAPEX total de 11,0 M€ hace que la inversión quede en el umbral de viabilidad, con un VAN de -128.678 € y una TIR del 7,82 %.

El Modelo 6, basado en almacenamiento térmico de frío, alcanza un coste anual de 7.918.526 €, con un ahorro energético de 985.350 € (11,07 %). Aunque este ahorro es inferior al del BESS, el CTES requiere un CAPEX menor, 7,4 M€, no necesita reemplazo dentro del horizonte de análisis y libera capacidad eléctrica asociada al sistema de refrigeración, valorada en 198.476 €/año. Al incorporar este valor adicional, el beneficio bruto anual asciende a 1.183.826 €, con un VAN de 1.990.431 €, una TIR del 12,05 % y un payback simple de 7 años.

Por último, el Modelo 7 combina la batería y el CTES en un único sistema coordinado y alcanza el menor coste anual de todos los modelos, 7.302.797 €, con un ahorro energético de 1.601.078 € (17,98 %) y un beneficio bruto de 1.799.489 €/año al incluir el valor de la capacidad IT liberada. Sin embargo, su CAPEX conjunto de 13,4 M€ reduce la rentabilidad frente al CTES en solitario, con un VAN de 422.196 € y una TIR del 8,49 %. El análisis incremental muestra que añadir el CTES sobre una batería ya instalada sí resulta rentable (VAN incremental de 550.874 €), mientras que añadir la batería sobre un CTES existente no lo es (VAN incremental de -1.568.235 €).

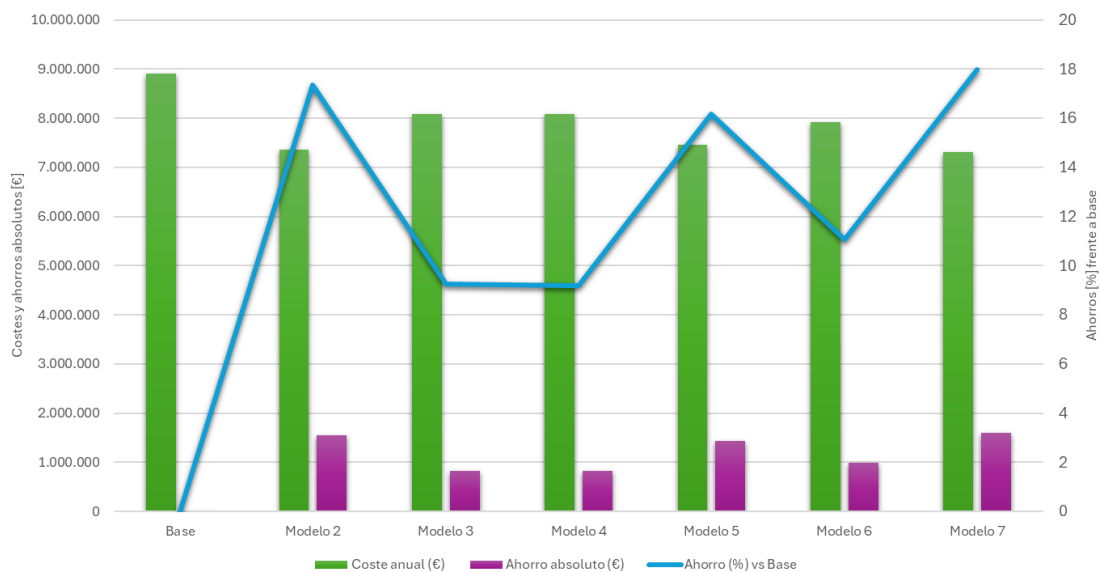


Figura 1. Gráfica de costes vs ahorros.

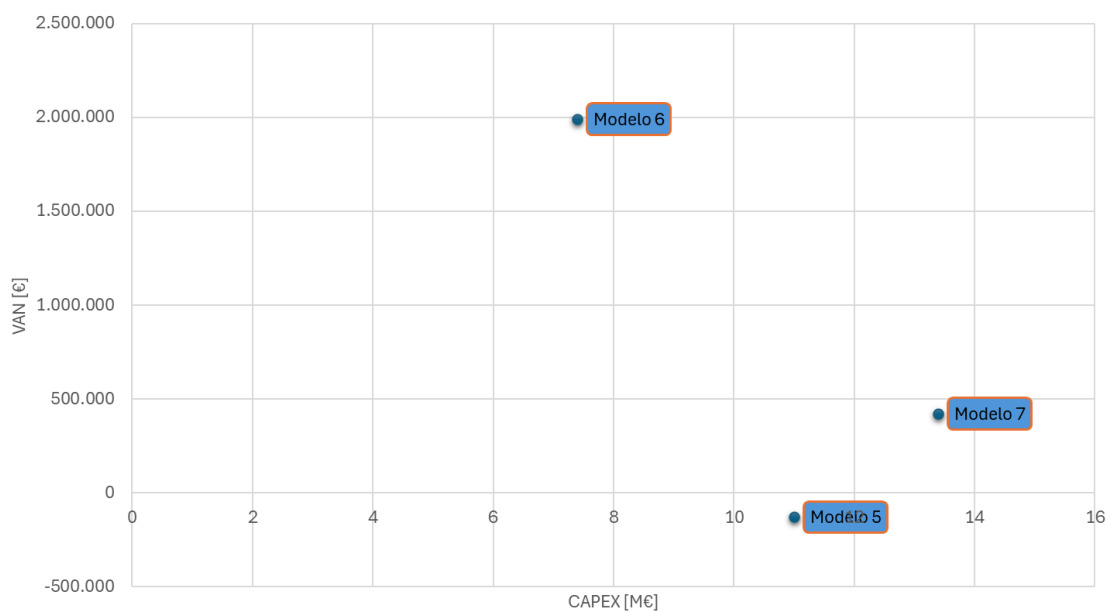


Figura 2. Gráfica de relación entre CAPEX y VAN de alternativas de inversión.

Tabla 1. Tabla resumen resultados modelos

Modelo	Medida incorporada	Coste anual (€)	Ahorro (€/año)	Ahorro (%)	CAPEX (€)	VAN 8 %, 15 a (€)	TIR (%)
M1	Caso base	8.903.875	—	—	0	—	—
M2	Training anual	7.358.840	1.545.035	17,35 %	0	—	—
M3	Training diario	8.080.644	823.231	9,25 %	0	—	—
M4	+Refrigeración amoníaco	8.081.904	821.972	9,23 %	4.500.000	n/a	n/a
M5	+Batería BESS	7.464.296	1.439.579	16,17 %	11.000.000	-128.678	7,82 %
M6	+ CTES	7.918.526	1.183.826*	11,07 % **	7.400.000	1.990.431	12,05 %
M7	Híbrido BESS+CTES	7.302.797	1.799.489*	17,98 % **	13.400.000	422.196	8,49 %

*Beneficio bruto incluye valor de capacidad IT liberada. **% solo sobre ahorro energético puro.

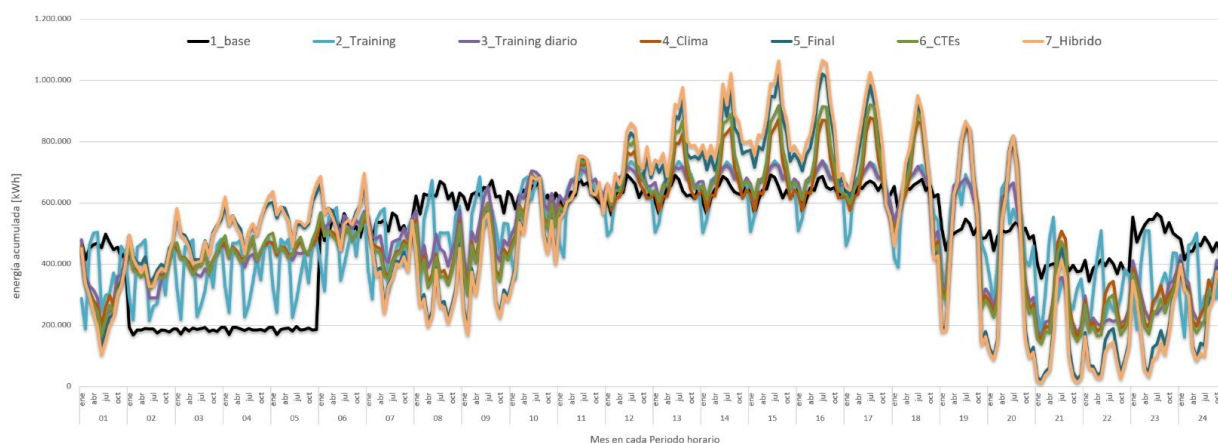


Figura 3. Training original vs. training optimizado (kWh) por mes en periodo horario de cada modelo.

5. Conclusiones

La principal conclusión del trabajo es que la flexibilidad de la carga de entrenamiento constituye la medida más inmediata y eficiente, ya que permite obtener ahorros

relevantes sin necesidad de inversión adicional. En particular, la flexibilidad diaria del Modelo 3 representa una estrategia realista y de bajo riesgo para reducir la factura eléctrica de un centro de datos dedicado a IA.

Entre las alternativas que requieren inversión, el sistema CTES resulta la opción más atractiva en el escenario base. Aunque no consigue un ahorro energético tan alto como el BESS, combina un menor CAPEX, ausencia de reemplazo y un valor adicional asociado a la capacidad eléctrica liberada en refrigeración. El BESS queda cerca del umbral de viabilidad y su rentabilidad dependerá de la evolución futura del precio eléctrico, de la volatilidad del mercado y del coste de las baterías. El análisis de sensibilidad muestra que bastaría un crecimiento del precio de la energía del 0,2 % anual, o una reducción del CAPEX de la batería del 1,3 %, para que el VAN pasara a ser positivo. Por su parte, el sistema híbrido del Modelo 7 logra el mayor ahorro absoluto y mantiene un VAN positivo, pero su rentabilidad por euro invertido es inferior a la del CTES en solitario, por lo que ambas tecnologías se plantean como alternativas de inversión igualmente válidas, cada una preferible según el criterio de decisión del promotor: el CTES maximiza el retorno por euro invertido, mientras que el híbrido maximiza el ahorro absoluto, con un margen de seguridad más ajustado frente a variaciones de los parámetros clave. El análisis incremental indica además que el orden de inversión importa: ampliar con un tanque CTES una instalación que ya cuenta con batería resulta rentable, mientras que el camino inverso no lo es.

6. Referencias

Las referencias empleadas incluyen documentación técnica y académica sobre centros de datos, consumo energético de la inteligencia artificial, mercados eléctricos, refrigeración industrial, almacenamiento en batería, almacenamiento térmico y herramientas de optimización matemática. Entre ellas destacan informes de la International Energy Agency, Uptime Institute, OMIE, ASHRAE, NREL, Lazard y documentación técnica de Pyomo y HiGHS.

ECONOMIC ANALYSIS OF THE ENERGY SUPPLY IN DATA CENTERS DEDICATED TO ARTIFICIAL INTELLIGENCE

Author: Carlota Otero Galeano

Director: Javier Ponce Cintas

Collaborating Entity: ICAI – Universidad Pontificia Comillas

PROJECT SUMMARY

This Final Degree Project studies the optimization of the electricity supply of a Data Center with 20 MW of IT power dedicated to artificial intelligence, connected to the Spanish wholesale electricity market (OMIE) and evaluated with hourly data for the year 2025. To this end, seven optimization models are formulated in Python with Pyomo and solved with HiGHS, incrementally incorporating different degrees of freedom: the possibility of shifting the training workload over time, a cooling system modeled from the outdoor temperature, a large-scale battery that exploits the hourly volatility of the market, a cold thermal energy storage (CTES) system that performs that same arbitrage on the cooling demand and, finally, a hybrid system that combines both storage technologies. Each model is assessed through economic indicators such as NPV, IRR and payback period, together with a sensitivity analysis of the most uncertain parameters.

Keywords: Data Centers, Artificial intelligence, Energy optimization, OMIE electricity market, Demand flexibility, Battery energy storage, CTES, Ammonia refrigeration.

1. Introduction

In just a few years, artificial intelligence has become a central piece of the digital economy, and the Data Centers that support it run practically 24/7 at full power. The latest reports already place their consumption at around 1.5 % of the world's electricity, and everything suggests that this figure will keep growing as large-scale models become

widespread. For the operator, the consequence is that the electricity bill has become one of the heaviest items in its operating account.

The Spanish wholesale electricity market, managed by OMIE, features highly variable hourly prices, and that variability can work both for and against the operator. If part of the load can wait, as is the case with certain training jobs, or if storage is available, consumption can be shifted towards cheap hours and purchases can be reduced during expensive ones. This leads to the main question of this project: how much can actually be saved with this type of strategy, and at what point does the required investment pay off.

2. Project definition

The goal of this project is to build several optimization models of the electricity supply of an AI Data Center and, based on them, measure the savings and profitability of each strategy. The process is organized in three blocks. The first prepares the base case, with OMIE hourly prices and the demand profile over the 8,760 hours of the year. The second formulates seven optimization models, each one adding a degree of freedom with respect to the previous one. Models 1 to 4 are stated as continuous linear problems, while Models 5 to 7 incorporate binary variables to prevent the simultaneous charging and discharging of the storage systems, and are therefore solved as mixed-integer linear problems (MILP). The third block evaluates each case economically through profitability indicators and sensitivity analyses.

3. Description of the model/system/tool

The modeled Data Center has 20 MW of IT power and a PUE of 1.3, which results in a total power of 26 MW. Its load is split between two types of task: training, which is flexible and accounts for 60 % of IT consumption, and inference, which is rigid and accounts for the remaining 40 %. Seven models have been developed on top of this base case:

- Model 1: reproduces the operation without any management and sets the reference cost.
- Model 2: allows the training load to be shifted across the whole year, a theoretical bound that is not realistic in practice.
- Model 3: restricts that flexibility to within each day, which is operationally feasible.
- Model 4: replaces the dataset cooling load with a model calibrated against the outdoor temperature.
- Model 5: adds a 20 MWh BESS battery that performs hourly arbitrage in the electricity market.
- Model 6: replaces that battery with a cold thermal energy storage (CTES) system, which shifts the cooling demand to the cheapest hours with a lower CAPEX and no replacement needed within the analysis horizon.
- Model 7: combines the BESS battery and the CTES tank on the same basis, coordinating both storage systems together with the training flexibility; it achieves the largest absolute savings of all the models, although its return per euro invested is lower than that of the CTES alone.

All the models have been implemented in Python with Pyomo and solved with the HiGHS solver.

4. Results

The annual cost of the base case amounts to €8,903,875. Model 2, with annual training flexibility, reduces the cost to €7,358,840, which represents savings of €1,545,035 (17.35 %). This result is a theoretical bound, since it allows the training load to be shifted freely across the entire year. When that flexibility is restricted to within each day, Model 3 obtains a cost of €8,080,644, with savings of €823,231 (9.25 %) and no capital investment required.

Model 4, which incorporates cooling dependent on the outdoor temperature, yields an annual cost of €8,081,904, practically identical to that of Model 3. This indicates that the climate model consistently reproduces the original cooling profile, but does not by itself generate any significant additional savings. Model 5, with the BESS battery, lowers the annual cost to €7,464,296, with savings of €1,439,579 (16.17 %). However, its total CAPEX of €11.0 M leaves the investment at the threshold of viability, with an NPV of –€128,678 and an IRR of 7.82 %.

Model 6, based on cold thermal storage, reaches an annual cost of €7,918,526, with energy savings of €985,350 (11.07 %). Although these savings are lower than those of the BESS, the CTES requires a smaller CAPEX, €7.4 M, needs no replacement within the analysis horizon and frees up electrical capacity associated with the cooling system, valued at €198,476/year. When this additional value is included, the gross annual benefit rises to €1,183,826, with an NPV of €1,990,431, an IRR of 12.05 % and a simple payback of 7 years.

Finally, Model 7 combines the battery and the CTES in a single coordinated system and achieves the lowest annual cost of all the models, €7,302,797, with energy savings of €1,601,078 (17.98 %) and a gross benefit of €1,799,489/year once the value of the released IT capacity is included. However, its combined CAPEX of €13.4 M reduces its profitability compared with the CTES alone, with an NPV of €422,196 and an IRR of 8.49 %. The incremental analysis shows that adding a CTES to an already installed battery is indeed profitable (incremental NPV of €550,874), whereas adding a battery to an existing CTES is not (incremental NPV of –€1,568,235).

Figure 1 summarizes this comparison: the annual cost falls steadily as degrees of freedom are added, while the savings grow from Model 1 to Model 7.

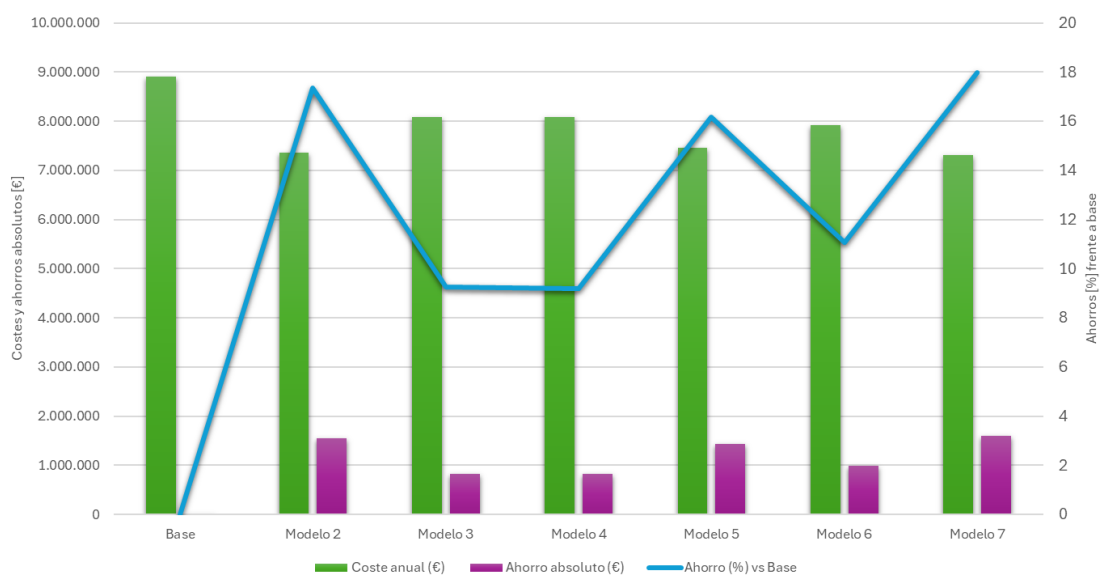


Figure 1. Annual costs vs. savings by model.

Figure 2 places each investment alternative on the CAPEX–NPV plane and makes the trade-off visible at a glance: the CTES (M6) clearly dominates in return per euro invested, the hybrid (M7) achieves the largest absolute savings at a higher capital cost, and the standalone BESS (M5) sits right at the viability threshold.

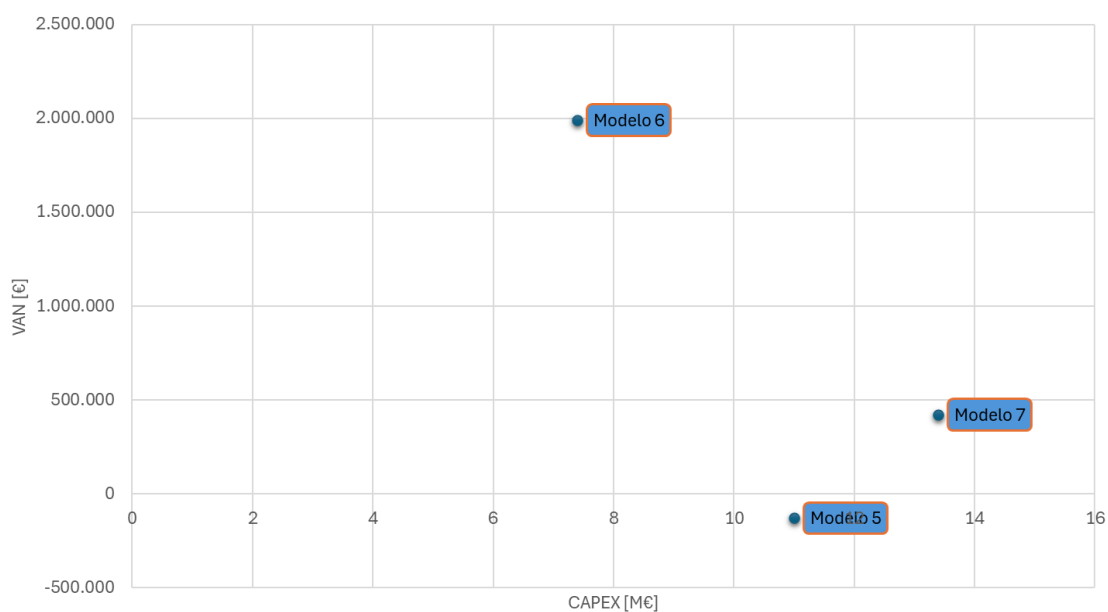


Figure 2. CAPEX vs. NPV of the investment alternatives.

Table 1 gathers the annual costs, savings, CAPEX, NPV and IRR of the seven models.

Table 1. Summary of the model results.

Model	Measure added	Annual cost (€)	Savings (€/yr)	Savings (%)	CAPEX (€)	NPV 8 %, 15 y (€)	IRR (%)
M1	Base case	8,903,875	—	—	0	—	—
M2	Annual training flexibility	7,358,840	1,545,035	17.35 %	0	—	—
M3	Daily training flexibility	8,080,644	823,231	9.25 %	0	—	—
M4	+ Ammonia cooling	8,081,904	821,972	9.23 %	4,500,000	n/a	n/a
M5	+ BESS battery	7,464,296	1,439,579	16.17 %	11,000,000	−128,678	7.82 %
M6	+ CTES	7,918,526	1,183,826*	11.07 %**	7,400,000	1,990,431	12.05 %
M7	BESS+CTES hybrid	7,302,797	1,799,489*	17.98 %**	13,400,000	422,196	8.49 %

*Gross benefit includes the value of the released IT capacity. **Percentage refers to pure energy savings only.

Finally, Figure 3 illustrates the operational mechanism behind these savings: month after month, the optimizer systematically displaces the training load from expensive to cheap hours while always completing the same total amount of work.

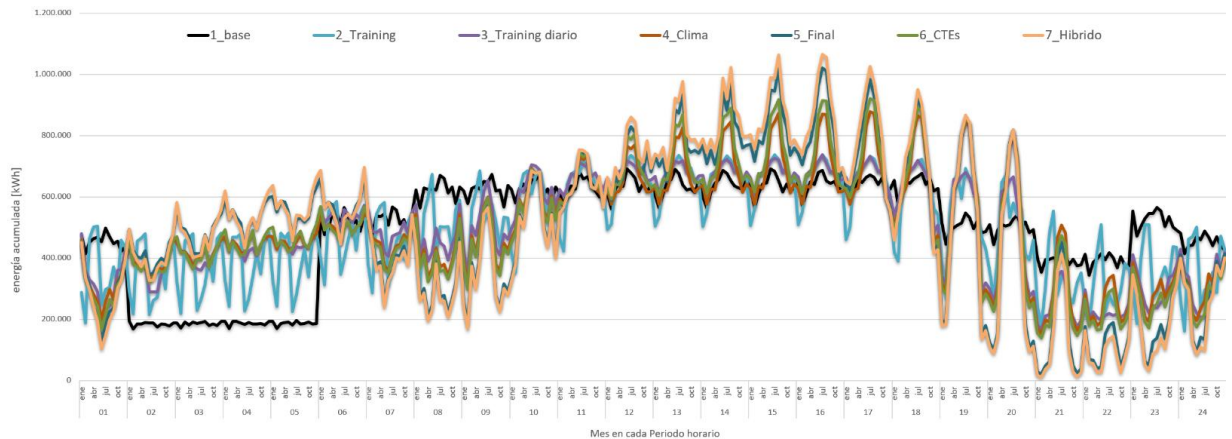


Figure 3. Original vs. optimized training (kWh) per month and hourly period for each model.

In short, the daily flexibility of the training load emerges as the immediate, zero-cost measure; the CTES as the most profitable investment; and the BESS+CTES hybrid as

the option that maximizes absolute savings: an escalation path in which each step adds value on top of the previous one.

5. Conclusions

The main conclusion of this project is that the flexibility of the training load is the most immediate and efficient measure, since it delivers substantial savings without any additional investment. In particular, the daily flexibility of Model 3 represents a realistic, low-risk strategy to reduce the electricity bill of an AI-dedicated Data Center.

Among the alternatives that require investment, the CTES system turns out to be the most attractive option in the base scenario. Although it does not achieve energy savings as high as those of the BESS, it combines a lower CAPEX, no replacement needs and an additional value associated with the electrical capacity released from the cooling system. The BESS remains close to the viability threshold, and its profitability will depend on the future evolution of electricity prices, market volatility and battery costs. The sensitivity analysis shows that an annual energy price growth of just 0.2 %, or a 1.3 % reduction in the battery CAPEX, would be enough to turn the NPV positive. The hybrid system of Model 7 achieves the largest absolute savings and keeps a positive NPV, but its return per euro invested is lower than that of the CTES alone, so both technologies stand as equally valid investment alternatives, each one preferable depending on the decision criterion of the promoter: the CTES maximizes the return per euro invested, while the hybrid maximizes absolute savings with a tighter safety margin against variations in the key parameters. The incremental analysis also indicates that the order of investment matters: expanding a facility that already has a battery with a CTES tank is profitable, whereas the reverse path is not.

6. References

The references used include technical and academic literature on Data Centers, the energy consumption of artificial intelligence, electricity markets, industrial refrigeration, battery storage, thermal storage and mathematical optimization tools. They notably

include reports by the International Energy Agency, Uptime Institute, OMIE, ASHRAE, NREL and Lazard, as well as the technical documentation of Pyomo and HiGHS.

Índice de la memoria

Capítulo 1. Introducción	26
1.1 Motivación del proyecto	26
1.2 Estructura de la memoria	27
Capítulo 2. Descripción de las Tecnologías.....	29
2.1 Centros de datos para inteligencia artificial	29
2.1.1 Qué es un CPD dedicado a IA y en qué se diferencia del tradicional	29
2.1.2 Cargas de trabajo: entrenamiento vs inferencia.....	30
2.1.3 Perfil de consumo y flexibilidad de la demanda	31
2.1.4 PUE como métrica de eficiencia energética	32
2.2 El mercado eléctrico mayorista: precios horarios OMIE.....	33
2.2.1 Funcionamiento del mercado spot	33
2.2.2 Volatilidad del precio y oportunidades de arbitraje.....	33
2.3 Sistemas de refrigeración industrial.....	34
2.3.1 Necesidad de refrigeración en CPDs.....	34
2.3.2 Ciclo de refrigeración por compresión.....	34
2.3.3 Refrigeración por amoníaco: COP y rendimiento eléctrico-térmico	36
2.4 Almacenamiento de energía en batería (BESS).....	37
2.4.1 Tecnología Li-ion: principio de funcionamiento	37
2.4.2 Ciclo de carga/descarga, SOC y eficiencias	38
2.4.3 Degradación y vida útil.....	38
2.4.4 CAPEX y tendencia de costes.....	39
2.5 Optimización matemática con Pyomo	39
Capítulo 3. Estado de la Cuestión	41
3.1 Evolución de la gestión energética en centros de datos	41
3.2 Demand response y flexibilidad de carga en CPDs de IA	41
3.3 Almacenamiento en batería y arbitraje energético.....	42
3.4 Refrigeración, clima y almacenamiento térmico	44
3.5 Análisis económico de inversiones energéticas	45
3.6 Hueco identificado y aportación de este TFG.....	45

Capítulo 4. Definición del Trabajo	47
4.1 Justificación.....	47
4.2 Objetivos	48
4.3 Metodología	49
4.4 Planificación y estimación económica	50
4.5 Alineación con los Objetivos de Desarrollo Sostenible (ODS)	52
Capítulo 5. Sistema/Modelo Desarrollado	54
5.1 Descripción del caso base: el centro de datos modelado	54
5.2 Datos de entrada: precios OMIE y perfil de demanda	55
5.3 Modelo 1: Caso base sin optimización.....	56
5.3.1 Descripción	56
5.3.2 Formulación matemática	56
5.4 Modelo 2: Flexibilidad de training anual.....	58
5.4.1 Descripción y motivación.....	58
5.4.2 Formulación matemática	58
5.5 Modelo 3: Flexibilidad de training diaria.....	60
5.5.1 Descripción y motivación.....	60
5.5.2 Formulación matemática	60
5.5.3 Resultados	61
5.6 Modelo 4: Modelo climático y refrigeración por amoníaco	62
5.6.1 Descripción y motivación.....	63
5.6.2 Modelo de demanda de refrigeración	63
5.6.3 Función objetivo.....	64
5.6.4 Resultados	64
5.7 Modelo 5: Sistema completo con almacenamiento en batería	65
5.7.1 Descripción y motivación.....	65
5.7.2 Parámetros de la batería.....	66
5.7.3 Variables adicionales.....	66
5.7.4 Formulación matemática completa.....	66
5.7.5 Resultados	68
5.8 Modelo 6: Sistema CTES con almacenamiento térmico de frío	70
5.8.1 Descripción y motivación.....	70
5.8.2 Parámetros del sistema	72

5.8.3	Formulación matemática	73
5.8.4	Resultados	75
5.9	Modelo 7: Sistema híbrido con batería BESS y almacenamiento térmico CTES.....	77
5.9.1	Descripción y motivación.....	77
5.9.2	Parámetros del sistema	78
5.9.3	Variables adicionales.....	80
5.9.4	Formulación matemática completa.....	80
5.9.5	Resultados	81
Capítulo 6.	Análisis de Resultados.....	84
6.1	Comparación de costes entre modelos	84
6.1.1	Costes anuales y ahorros	84
6.1.2	Análisis de los resultados por modelo.....	85
6.2	Análisis económico de la inversión.....	87
6.2.1	Hipótesis de CAPEX y OPEX.....	87
6.2.2	VAN, TIR y periodo de retorno	88
6.2.3	Comparativa de tecnologías de almacenamiento: BESS, CTES e Híbrido	90
6.3	Análisis de sensibilidad.....	92
6.3.1	Sensibilidad del Modelo 5 al WACC, al crecimiento del precio de la energía y al CAPEX	93
6.3.2	Sensibilidad al coste del tanque y al beneficio total anual	94
6.3.3	Sensibilidad al WACC y al crecimiento del precio de la energía.....	94
6.3.4	Sensibilidad al valor de la capacidad IT liberada.....	95
6.3.5	Sensibilidad al beneficio total y al CAPEX del M6	97
Capítulo 7.	Conclusiones y Trabajos Futuros.....	100
7.1	Conclusiones	100
7.1.1	Primera conclusión: la flexibilidad de la demanda de training es la palanca de mayor valor y menor coste.	100
7.1.2	Segunda conclusión: la refrigeración por amoníaco aporta valor técnico pero no económico incremental sobre M3.	102
7.1.3	Tercera conclusión: el BESS se encuentra en el umbral de viabilidad y su rentabilidad depende críticamente de los parámetros adoptados.	103
7.1.4	Cuarta conclusión: el CTES es la alternativa más rentable entre las que requieren inversión.	103

7.1.5 Quinta conclusión: el híbrido BESS+CTES maximiza el ahorro absoluto con un margen de seguridad más ajustado, y el orden de inversión importa.....	104
7.1.6 Síntesis general.	105
7.2 Trabajos futuros	107
7.2.1 Precios intradiarios y mercados de servicios de ajuste.	107
7.2.2 Horizonte de optimización rodante.	107
7.2.3 Integración de energía solar fotovoltaica.	107
7.2.4 Optimización multiobjetivo con emisiones de CO ₂	107
7.2.5 Validación con datos reales de operación.	108
7.2.6 Análisis de degradación del BESS con modelo de envejecimiento.	108
Capítulo 8. Bibliografía.....	109
 ANEXO I 111	
8.1 Modelo 1: Caso base sin optimización.....	111
8.2 Modelo 2: Flexibilidad de training anual.....	114
8.3 Modelo 3: Flexibilidad de training diaria.....	119
8.4 Modelo 4: Modelo climático y refrigeración por amoníaco	125
8.5 Modelo 5: Sistema completo con almacenamiento en batería.....	131
8.6 Modelo 6: Sistema CTES con almacenamiento térmico de frío.....	138
8.7 Modelo 7: Sistema híbrido con batería BESS y almacenamiento térmico CTES.....	148
8.8 Análisis económico	158
8.9 Archivos de datos y resultados complementarios	186

Índice de tablas

Tabla 1. Tabla resumen resultados modelos.....	IX
Tabla 2. Parámetros técnicos del centro de datos modelado.	54
Tabla 3. Parámetros del sistema BESS en el Modelo 5.....	66
Tabla 4. Parámetros del sistema CTES en el Modelo 6.	72
Tabla 5. Parámetros del sistema BESS en el Modelo 7 (idénticos al Modelo 5).	78
Tabla 6. Parámetros del sistema CTES en el Modelo 7 (idénticos al Modelo 6).	79
Tabla 7. Comparativa de costes anuales, ahorros y CAPEX por modelo.	85
Tabla 8. Desglose de CAPEX por componente.....	87
Tabla 9. Indicadores de rentabilidad de los Modelos 5, 6 y 7.	88
Tabla 10. Sensibilidad del VAN de M6 (€) al coste unitario del tanque y al beneficio total anual.....	94
Tabla 11. Sensibilidad del VAN de M6 (€) al WACC y al crecimiento anual del precio de la energía.	95
Tabla 12. Sensibilidad del VAN de M6 (€) al precio del compute y al factor de uso de la capacidad IT liberada.....	95
Tabla 13. Sensibilidad del VAN de M6 (€) al beneficio total y al CAPEX.....	97
Tabla 14. Training adicional comprable con el ahorro anual de cada modelo.	101
Tabla 15. Resumen de resultados y análisis de inversión de los siete modelos.	105

Índice de ilustraciones

Figura 1. Gráfica de costes vs ahorros.....	VIII
Figura 2. Gráfica de relación entre CAPEX y VAN de alternativas de inversión.....	VIII
Figura 3. Training original vs. training optimizado (kWh) por mes en periodo horario de cada modelo.....	IX
Figura 4. Esquema simplificado del ciclo de refrigeración por compresión de vapor.	35
Figura 5. Esquema simplificado de un sistema de refrigeración por compresión con amoníaco.....	36
Figura 6. Diagrama de bloques del funcionamiento de Pyomo y HiGHS.....	40
Figura 7. GANTT del proyecto	51
Figura 8 – Diagrama base Modelo 1	56
Figura 9 – Diagrama base Modelo 2	58
Figura 10 – Diagrama base Modelo 3	60
Figura 11. Energía de training diaria Modelo 2 frente a Modelo 3.	62
Figura 12 – Diagrama base Modelo 4	62
Figura 13 – Diagrama base Modelo 5	65
Figura 14. Energía gestionada por la batería BESS en el Modelo 5	69
Figura 15 – Diagrama base Modelo 6	70
Figura 16. Capacidad eléctrica IT liberada por el CTES del Modelo 6	76
Figura 17 – Diagrama base Modelo 7	77
Figura 18. Energía gestionada por la batería BESS en el Modelo 7	83
Figura 19. Capacidad eléctrica de IT liberada en el Modelo 7.....	83
Figura 20 – Diagrama base Análisis de resultados.....	84
Figura 21. Coste horario optimizado agregado por modelo	90
Figura 22. Decisión de inversión entre BESS, CTES e híbrido: CAPEX frente a VAN	91
Figura 23. Training original vs. training optimizado (kWh) por mes en periodo horario de cada modelo.....	102
Figura 24. Evolución del beneficio acumulado de los modelos de inversión	106

Índice de Ecuaciones

Ecuación 1 – Definición de PUE	32
Ecuación 2 – Definición de COP.....	35
Ecuación 3 – Definición de SOC en función del tiempo.....	38
Ecuación 4 – Energía adquirida de red en periodo horario	57
Ecuación 5 – Función objetivo utilizada	57
Ecuación 6 – Demanda total horaria.....	59
Ecuación 7 – Energía anual de training conservada	59
Ecuación 8 – Límite de potencia máxima	59
Ecuación 9 – Función objetivo actualizada	59
Ecuación 10 – Restricción de energía diaria	61
Ecuación 11 – Consumo eléctrico de refrigeración dependiente de la temperatura.....	63
Ecuación 12 – Demanda total horaria con sistema de amoníaco	64
Ecuación 13 – Balance eléctrico con batería	67
Ecuación 14 – Evolución del SOC de la batería.....	67
Ecuación 15 – Límites del SOC de la batería	67
Ecuación 16 – Límite de potencia de carga de la batería	67
Ecuación 17 – Límite de potencia de descarga de la batería	67
Ecuación 18 – Condición cíclica anual de la batería	68
Ecuación 19 – Función objetivo del Modelo 5	68
Ecuación 20 – Balance eléctrico del Modelo 6 con CTES.....	73
Ecuación 21 – Restricción de consumo no negativo del chiller	74
Ecuación 22 – Evolución del SOC del tanque térmico	74
Ecuación 23 – Límites del SOC del tanque térmico en unidades eléctricas equivalentes... 74	
Ecuación 24 – Límites de potencia y no simultaneidad del CTES.....	74
Ecuación 25 – Condición cíclica anual del tanque térmico	75
Ecuación 26 – Función objetivo del Modelo 6.....	75
Ecuación 27 – Balance eléctrico del Modelo 7 con BESS y CTES	80
Ecuación 28 – Función objetivo del Modelo 7	81

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN DEL PROYECTO

Con todas las noticias sobre inteligencia artificial que circulan en los últimos años, resulta difícil no preguntarse dónde se almacenan realmente todos esos datos y a qué coste energético. Esta reflexión condujo a enfocar el trabajo en los centros de datos, un sector clave dentro del proceso de digitalización y del desarrollo de la inteligencia artificial. Está claro que los centros de datos serán una de las infraestructuras críticas del próximo ciclo económico. Además, se trata de un sector con importantes oportunidades profesionales para los ingenieros de tecnologías industriales.

Los centros de datos consumen ya entre el 1 % y el 2 % de la electricidad mundial [1], y la cifra crecerá conforme la IA se extienda. Para un centro de datos de IA, que opera próximo a su carga nominal de forma continua, la factura energética representa una de sus partidas más significativas.

Además, se consideró especialmente interesante abordar un ámbito todavía poco desarrollado desde el punto de vista académico. Aunque existe información sobre centros de datos, eficiencia energética o mercados eléctricos por separado, la bibliografía específica sobre la gestión energética de centros de datos dedicados a inteligencia artificial es más limitada. Esto hizo que la fase inicial de documentación tuviera un peso importante dentro del trabajo, ya que fue necesario estudiar conceptos muy diversos, como la arquitectura del hardware, los perfiles de carga asociados al entrenamiento de modelos, la refrigeración de la instalación o el funcionamiento de los mercados eléctricos.

En España, los precios horarios del mercado mayorista (OMIE) muestran una volatilidad considerable ya que dentro de un mismo día hay horas valle y horas punta con diferencias

relevantes. Para un operador con capacidad de gestionar parte de su demanda o de almacenar energía es una oportunidad.

¿Hasta qué punto se puede reducir el coste eléctrico anual de un centro de datos mediante una gestión activa de su consumo y en qué condiciones resulta justificable invertir en los activos que hacen posible dicha gestión, como la flexibilidad de carga, la optimización de la refrigeración o el almacenamiento energético? Para responder a esta cuestión, se han desarrollado distintos modelos de optimización de complejidad creciente. Estos modelos parten de un caso base sin gestión activa y avanzan progresivamente hacia formulaciones más completas, incorporando la flexibilidad de determinadas cargas, la influencia de la temperatura ambiente sobre la refrigeración y el uso de sistemas de almacenamiento. El último modelo analiza una alternativa al almacenamiento electroquímico convencional mediante baterías BESS: un sistema de almacenamiento térmico de frío, o CTES. Esta resulta especialmente interesante porque, frente a una batería eléctrica, permite representar de forma más directa la relación entre la temperatura ambiente, la demanda de refrigeración y el rendimiento energético del sistema.

1.2 ESTRUCTURA DE LA MEMORIA

La memoria se organiza en ocho capítulos. El *Capítulo 2.* describe las tecnologías y conceptos en los que se apoya este trabajo como los CD dedicados a IA y sus particularidades energéticas, el mercado eléctrico horario, los sistemas de refrigeración industrial y el almacenamiento en baterías. El *Capítulo 3.* revisa el estado del arte sobre la gestión energética y como implementarla a los CD, e identifica el hueco que quiere cubrir el proyecto. El *Capítulo 4.* define los objetivos, la metodología, la planificación y la alineación con los Objetivos de Desarrollo Sostenible. El *Capítulo 5.* desarrolla los siete modelos de optimización, desde la formulación matemática hasta los parámetros empleados. El Capítulo 6. expone el análisis de resultados, con la comparativa de costes, el análisis de inversión y

los de sensibilidad. Y por último, el *Capítulo 7*. recoge las conclusiones y propone líneas futuras. El *Capítulo 8*. contiene las referencias bibliográficas.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

2.1 CENTROS DE DATOS PARA INTELIGENCIA ARTIFICIAL

2.1.1 QUÉ ES UN CPD DEDICADO A IA Y EN QUÉ SE DIFERENCIA DEL TRADICIONAL

Un centro de procesamiento de datos (CPD o CD) es la instalación que reúne la infraestructura donde se almacena, procesa y se transmiten los datos necesarios para sostener los servicios digitales. En la práctica, estos centros permiten que funcionen muchos servicios digitales cotidianos, como las plataformas en la nube, las aplicaciones móviles o los buscadores.

Los CD dedicados a la IA presentan sus propios rasgos característicos que los diferencian de los CD tradicionales. Como referencia, este tipo de instalaciones puede consumir hasta diez veces más energía que un CD tradicional, y las previsiones apuntan a un crecimiento sostenido de la demanda eléctrica del sector en los próximos años [1].

La principal diferencia se encuentra en el hardware. Mientras que un centro de datos convencional puede estar orientado a tareas como almacenar información, alojar páginas web o gestionar bases de datos, un centro de datos de IA necesita ejecutar una cantidad muy elevada de operaciones matemáticas en paralelo. Para ello emplea procesadores especializados diseñados para realizar cálculos masivos de forma simultánea.

Un ejemplo sencillo sería el de una herramienta como ChatGPT. Para que el sistema pueda responder a una pregunta, primero ha sido necesario entrenar modelos con grandes volúmenes de información.

Estos procesadores especializados consumen mucha más energía que los procesadores convencionales. Mientras que un procesador de uso general puede situarse en torno a los 100 W o 200 W, una GPU de altas prestaciones puede demandar entre 700 W y 1.000 W durante su funcionamiento.

Desde el punto de vista energético, esta diferencia tiene tres consecuencias principales. En primer lugar, genera una demanda eléctrica elevada y sostenida. En segundo lugar, aumenta la cantidad de calor que debe evacuarse por unidad de superficie, ya que los equipos trabajan con una densidad de potencia mucho mayor. Y en tercer lugar, combina cargas con distintos grados de flexibilidad: algunas deben ejecutarse en tiempo real, como la respuesta a una consulta de un usuario, mientras que otras pueden desplazarse temporalmente, como parte del entrenamiento de un modelo. Esta característica abre la posibilidad de aplicar estrategias de gestión activa de la demanda que no siempre están disponibles en un centro de datos convencional.

2.1.2 CARGAS DE TRABAJO: ENTRENAMIENTO VS INFERENCIA

Las cargas de un CPD de IA se agrupan en dos categorías: el entrenamiento y la inferencia. Esta distinción resulta relevante porque cada una presenta un perfil energético y una flexibilidad temporal muy diferentes.

El entrenamiento es el proceso mediante el cual un modelo de inteligencia artificial aprende a realizar una tarea. De forma simplificada, puede entenderse como una fase de aprendizaje en la que el modelo recibe grandes cantidades de datos, identifica patrones y ajusta sus parámetros internos para mejorar sus respuestas. Por ejemplo, en el caso de un sistema como Claude, el entrenamiento permite que el modelo aprenda relaciones entre palabras, conceptos y contextos a partir de grandes volúmenes de texto e información.

Este proceso puede durar horas, días o incluso semanas, con los procesadores trabajando de forma intensiva. Desde el punto de vista energético, el entrenamiento presenta una característica interesante: parte de esta carga puede ser flexible en el tiempo. Es decir, si se

debe completar una determinada cantidad de cálculo en un día, no siempre es imprescindible ejecutarla en una hora concreta. Por ello, en el modelo se considera que parte del entrenamiento puede desplazarse hacia las horas en las que el precio eléctrico es menor, siempre que se respeten las restricciones de operación establecidas.

La inferencia, en cambio, es la fase en la que un modelo ya entrenado utiliza lo que ha aprendido para responder a una nueva entrada. En todos estos casos, el modelo no está aprendiendo desde cero, sino aplicando el conocimiento adquirido durante el entrenamiento. Como el usuario espera una respuesta inmediata, esta carga no puede desplazarse libremente a otra hora del día sin afectar a la calidad del servicio. Por este motivo, en el modelo la inferencia se representa como una carga rígida, es decir, una demanda que debe atenderse en el momento en que se produce.

Esta diferencia entre entrenamiento e inferencia es la base de los modelos de optimización desarrollados en el trabajo. El caso base considera una potencia IT instalada de 20 MW, repartida en un 60 % para entrenamiento y un 40 % para inferencia. Esto equivale a 12 MW asociados a entrenamiento y 8 MW asociados a inferencia.

2.1.3 PERFIL DE CONSUMO Y FLEXIBILIDAD DE LA DEMANDA

El consumo eléctrico de un centro de datos dedicado a inteligencia artificial no se limita a la carga computacional. Aunque la carga IT constituye el núcleo del consumo, la instalación necesita otros sistemas para poder operar de forma continua. La refrigeración, encargada de evacuar el calor generado por los equipos y mantenerlos dentro de su rango de temperatura adecuado, representa habitualmente el segundo mayor consumidor de energía, con un peso que oscila entre el 30 % y el 40 % del total en instalaciones convencionales [1].

Dentro de este sistema no solo consumen energía los equipos principales de producción de frío, sino también los elementos encargados de mover los fluidos por la instalación. En particular, las bombas de los circuitos hidráulicos tienen un papel relevante, ya que permiten transportar el agua fría o el fluido refrigerante hasta las zonas donde se encuentran los

equipos IT y devolverlo posteriormente al sistema de refrigeración. A ellas se suman los ventiladores y otros equipos auxiliares de circulación de aire. Estos consumos presentan normalmente un comportamiento más inercial y menos flexible que la carga computacional, por lo que deben considerarse dentro del balance energético del centro de datos.

Finalmente, existen otros consumos auxiliares del edificio, como la iluminación, los sistemas de alimentación ininterrumpida, los transformadores y otros equipos de soporte necesarios para garantizar la continuidad del servicio.

Gestionar este consumo de forma activa, mediante modelización climática, almacenamiento térmico o coordinación con los ciclos de la carga IT, es una de las palancas de eficiencia más relevantes en este tipo de instalaciones.

2.1.4 PUE COMO MÉTRICA DE EFICIENCIA ENERGÉTICA

El indicador estándar de eficiencia en centros de datos es el Power Usage Effectiveness (PUE), que se define como el cociente entre la potencia total de la instalación y la consumida exclusivamente por la carga IT:

$$PUE = \frac{P_{total}}{P_{IT}}$$

Ecuación 1 – Definición de PUE

Un PUE de 1,0 sería el caso ideal, en el que toda la energía consumida se destinaría al propósito propio del DC. En la práctica, los centros de datos modernos suelen situarse entre 1,2 y 1,4, mientras que las instalaciones más antiguas superan en ocasiones el valor de 1,5 [2]. En este trabajo se ha adoptado un PUE de 1,3 como valor de referencia, de modo que, sobre los 20 MW de potencia IT, la instalación demanda en total 26 MW.

2.2 EL MERCADO ELÉCTRICO MAYORISTA: PRECIOS HORARIOS OMIE

2.2.1 FUNCIONAMIENTO DEL MERCADO SPOT

En España, el precio de la electricidad en el mercado mayorista se fija mediante una subasta diaria gestionada por el Operador del Mercado Ibérico de Energía (OMIE).

En el mercado diario, productores y consumidores presentan ofertas de venta y de compra para cada una de las horas del día siguiente. El resultado de este proceso proporciona un precio horario de referencia, que sirve como señal para la planificación del consumo y para la operación de los agentes del sistema [3].

Además del mercado diario, existen mercados intradiarios y mecanismos posteriores de ajuste que permiten adaptar la programación inicial a las condiciones reales del sistema eléctrico hasta el tiempo real. Por este motivo, el coste final de la energía no depende únicamente del mercado diario, aunque el precio horario obtenido en este mercado constituye una referencia fundamental para analizar estrategias de gestión de la demanda [3].

Este funcionamiento tiene dos consecuencias directas para los grandes centros de datos. La primera es que los precios varían hora a hora a lo largo del día y del año, en función de las condiciones de oferta y demanda del sistema. La segunda es que un centro de datos dedicado a inteligencia artificial puede adaptar parte de su perfil de consumo a esa señal de precios, siempre que disponga de cargas flexibles, como el entrenamiento, o de sistemas de almacenamiento energético.

2.2.2 VOLATILIDAD DEL PRECIO Y OPORTUNIDADES DE ARBITRAJE

La entrada creciente de generación eólica y solar fotovoltaica en el mix eléctrico español genera una volatilidad considerable en los precios horarios de OMIE. En las horas de mucho sol o viento, la oferta renovable, con coste marginal casi nulo, desplaza a las tecnologías

convencionales y empuja los precios hacia valores muy bajos o incluso negativos [4]. En las horas punta con poca generación renovable, en cambio, el precio puede experimentar subidas pronunciadas.

Esta dispersión de precios es la principal fuente de valor de las estrategias de gestión activa de la demanda y del almacenamiento. Cuanto mayor es la diferencia entre el precio máximo y el mínimo de un día, mayor es el ahorro que puede capturarse desplazando el consumo o arbitrando con la batería entre las horas punta y las horas valle.

2.3 SISTEMAS DE REFRIGERACIÓN INDUSTRIAL

2.3.1 NECESIDAD DE REFRIGERACIÓN EN CPDs

El calor que generan los equipos debe evacuarse de forma continua para mantener los componentes dentro de su rango de temperatura operativa. En un CPD de IA, los servidores se agrupan físicamente en armarios o racks, que pueden contener múltiples GPU, procesadores, memoria, equipos de red y sistemas de alimentación. Por ello, cuando se habla de densidad de potencia por rack se hace referencia a la potencia eléctrica concentrada en uno de estos armarios. En este tipo de instalaciones, la densidad de potencia por rack puede estar en el orden de 50 kW o 100 kW, frente a los 5 kW o 10 kW de los centros convencionales [5], lo que obliga a disponer de sistemas de refrigeración de alta capacidad. La refrigeración deja de ser un consumo auxiliar menor y pasa a constituir una parte importante del balance energético de la instalación.

2.3.2 CICLO DE REFRIGERACIÓN POR COMPRESIÓN

El funcionamiento de la refrigeración mecánica se basa en el ciclo termodinámico de compresión de vapor. Un fluido refrigerante recorre un circuito cerrado en cuatro etapas. La evaporación, en la que el fluido absorbe el calor del espacio que se quiere enfriar y pasa de líquido a vapor. Luego por la compresión, en la que el compresor eleva la presión y la temperatura del vapor consumiendo electricidad. Después se pasa a la condensación, en la

que el vapor caliente cede su calor al exterior y vuelve a líquido. Y finalmente, la expansión, en la que una válvula reduce la presión y cierra el ciclo.

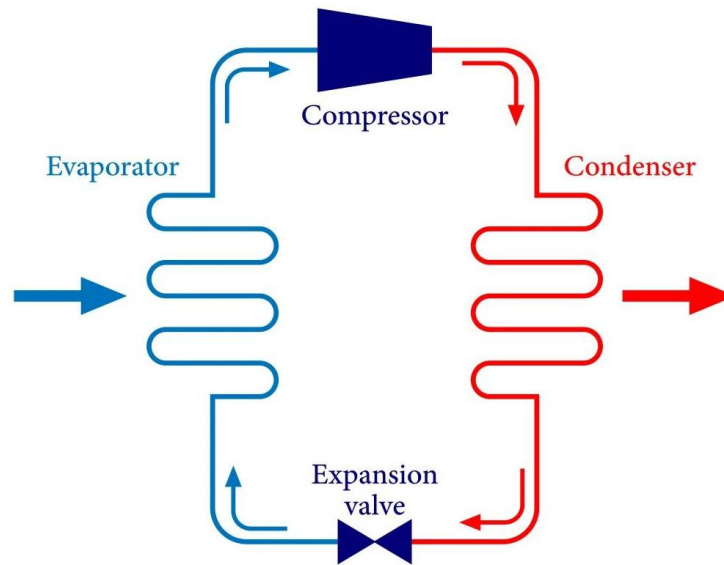


Figura 4. Esquema simplificado del ciclo de refrigeración por compresión de vapor.

La eficiencia del ciclo se mide con el coeficiente de funcionamiento o COP (Coefficient of Performance) que se define como el cociente entre el calor extraído del espacio refrigerado y el trabajo eléctrico que consume el compresor:

$$COP = \frac{Q_{frío}}{W_{elec}}$$

Ecuación 2 – Definición de COP

Un COP alto significa que el sistema extrae más calor por cada unidad de electricidad y por tanto es más eficiente.

2.3.3 REFRIGERACIÓN POR AMONIACO: COP Y RENDIMIENTO ELÉCTRICO-TÉRMICO

El amoníaco (NH_3) es uno de los refrigerantes más usados en instalaciones industriales que consumen una gran potencia. Su empleo se debe a sus buenas propiedades termodinámicas, su elevada eficiencia energética y su impacto prácticamente nulo sobre el calentamiento global, ya que presenta un potencial de calentamiento global muy bajo. No obstante, al tratarse de un fluido tóxico, su uso requiere medidas específicas de seguridad, detección de fugas y diseño adecuado de la instalación.

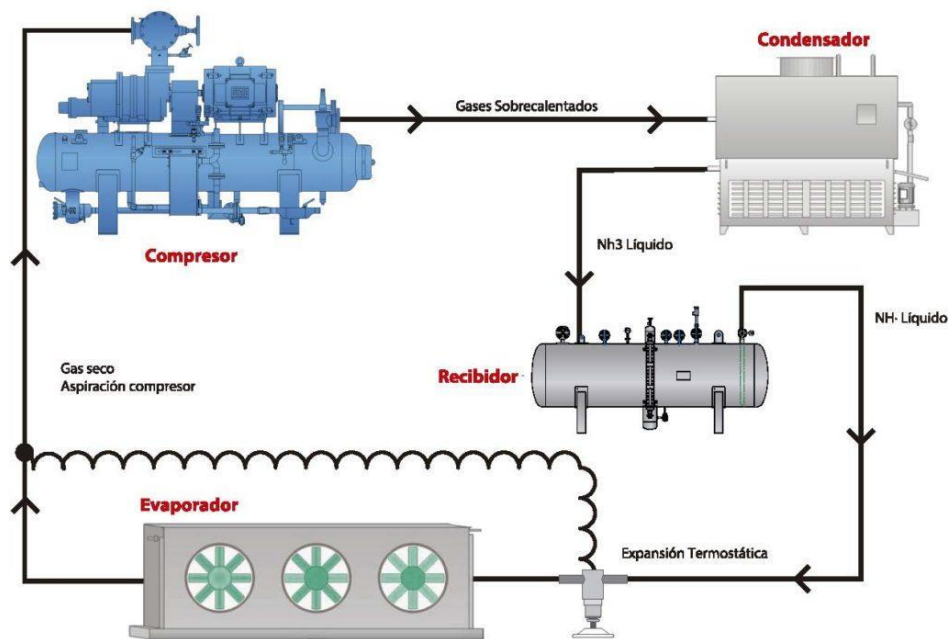


Figura 5. Esquema simplificado de un sistema de refrigeración por compresión con amoníaco.

El refrigerante circula por un ciclo de compresión de vapor, absorbiendo calor en el evaporador y cediéndolo al exterior en el condensador. En aplicaciones de gran potencia, como centros de datos, este ciclo permite producir frío de forma continua para evacuar el calor generado por los equipos IT.

En condiciones industriales reales, los sistemas de compresión con amoníaco alcanzan un COP eléctrico-térmico de entre 3 y 6 [6], según las temperaturas de operación y el ambiente. Estos valores muestran el interés del amoníaco en aplicaciones de refrigeración industrial de gran potencia, donde la eficiencia energética del sistema tiene un impacto directo sobre el consumo eléctrico total.

La demanda de refrigeración se modela como: una componente base, asociada al calor generado de forma continua por los equipos IT, y una componente variable dependiente de la temperatura exterior. De este modo, el modelo permite incorporar el efecto de las condiciones climáticas sobre la energía necesaria para refrigerar el centro de datos.

2.4 ALMACENAMIENTO DE ENERGÍA EN BATERÍA (BESS)

2.4.1 TECNOLOGÍA LI-ION: PRINCIPIO DE FUNCIONAMIENTO

Los sistemas de almacenamiento de energía en batería, conocidos por sus siglas inglesas BESS (Battery Energy Storage Systems), son ya una tecnología consolidada para gestionar la energía eléctrica en aplicaciones industriales y de red. La opción dominante es la batería de iones de litio (Li-ion), debido a su elevada densidad energética, su buen rendimiento de ciclo, la madurez de su cadena de suministro y la fuerte caída de costes registrada durante la última década [7].

Una celda Li-ion funciona por el movimiento reversible de iones de litio entre dos electrodos, el ánodo y el cátodo, a través de un electrolito. Durante la descarga, los iones migran del ánodo, normalmente formado por grafito, al cátodo, compuesto generalmente por sales de litio con metales de transición. Este movimiento genera una corriente externa que alimenta la carga conectada. Durante la carga, una corriente externa invierte el sentido del proceso.

2.4.2 CICLO DE CARGA/DESCARGA, SOC Y EFICIENCIAS

El estado de carga, o SOC (State of Charge), expresa la energía almacenada en un instante dado como una fracción de la capacidad nominal de la batería. Su valor se sitúa en el intervalo entre 0 y 1, equivalente a entre el 0 % y el 100 %. Un valor de 0 indica que la batería se encuentra totalmente descargada, mientras que un valor de 1 indica que se encuentra a plena capacidad.

La evolución del SOC a lo largo del tiempo viene dada por la siguiente ecuación de balance energético:

$$SOC_{(t)} = SOC_{(t-1)} + \eta_{carga} \cdot P_{carga_{(t)}} - \frac{P_{descarga_{(t)}}}{\eta_{descarga}}$$

Ecuación 3 – Definición de SOC en función del tiempo

donde η_{carga} y $\eta_{descarga}$ son las eficiencias de carga y descarga. En este trabajo se han adoptado eficiencias de 0,95 en ambos sentidos. Este valor se encuentra dentro del rango habitual empleado en modelos simplificados de sistemas BESS [8].

La batería opera con restricciones de potencia máxima de carga y descarga y dentro de un rango admisible de SOC. En aplicaciones reales, este rango suele limitarse para evitar operar de forma continuada en estados de carga extremos y preservar la vida útil del activo.

2.4.3 DEGRADACIÓN Y VIDA ÚTIL

Las baterías Li-ion pierden capacidad de manera gradual a lo largo de su vida útil, lo que se conoce como degradación. Esta degradación presenta dos componentes. La primera, de naturaleza cíclica, asociada al número y la profundidad de los ciclos de carga y descarga. La segunda, asociada al simple paso del tiempo y a las condiciones de almacenamiento. Tras diez años de operación intensiva en arbitraje, una batería puede haber perdido entre el 20 %

y el 30 % de su capacidad inicial. Esta pérdida recorta el ahorro alcanzable a lo largo de la vida del activo, por lo que resulta conveniente incorporarla al análisis económico.

2.4.4 CAPEX Y TENDENCIA DE COSTES

El coste de inversión de un BESS suele expresarse en euros por kilovatio-hora de capacidad instalada (€/kWh). Ha caído de forma muy marcada en la última década: de más de 1.000 €/kWh en 2010 a unos 200-300 €/kWh en instalaciones actuales, y se espera que siga bajando. En este trabajo se adopta un coste de referencia de 300 €/kWh sobre una capacidad de 20.000 kWh, lo que da un CAPEX de batería de 6.000.000 €. Este valor se utiliza como hipótesis base del análisis económico del Modelo 5.

2.5 OPTIMIZACIÓN MATEMÁTICA CON *PYOMO*

Los modelos de este trabajo se han implementado en Python con la librería Pyomo [9]. Esta herramienta permite formular problemas de optimización matemática de una forma parecida a como se escribirían en notación algebraica: se definen variables, restricciones y una función objetivo, y después el modelo se envía a un solver para obtener la solución óptima.

Para resolverlos se usa el solver HiGHS (High Performance Software for Linear Programming) [10]. HiGHS es una herramienta de código abierto preparada para resolver problemas lineales y lineales enteros mixtos, que son el tipo de problemas planteados en los modelos desarrollados. En concreto, los Modelos 1 a 4 se formulan como problemas lineales continuos, mientras que los Modelos 5 a 7 incorporan variables binarias para evitar la carga y descarga simultáneas de los sistemas de almacenamiento, por lo que se resuelven como problemas lineales enteros mixtos (MILP). La combinación de Pyomo y HiGHS permite trabajar con un horizonte anual horario, formado por 8.760 periodos, y con varias variables y restricciones asociadas a cada hora.

La Figura 5 resume el flujo seguido para resolver los modelos. En primer lugar, se incorporan las series de entrada, como los precios horarios de OMIE, la temperatura exterior y la demanda eléctrica. A partir de estos datos, Pyomo permite construir el modelo matemático, definiendo las variables de decisión, las restricciones operativas y la función objetivo. Después, el problema se traduce a un formato que puede resolver HiGHS. Finalmente, la solución obtenida proporciona los valores óptimos de las variables horarias, que se exportan a hojas de cálculo para revisar los resultados y realizar el análisis técnico y económico.

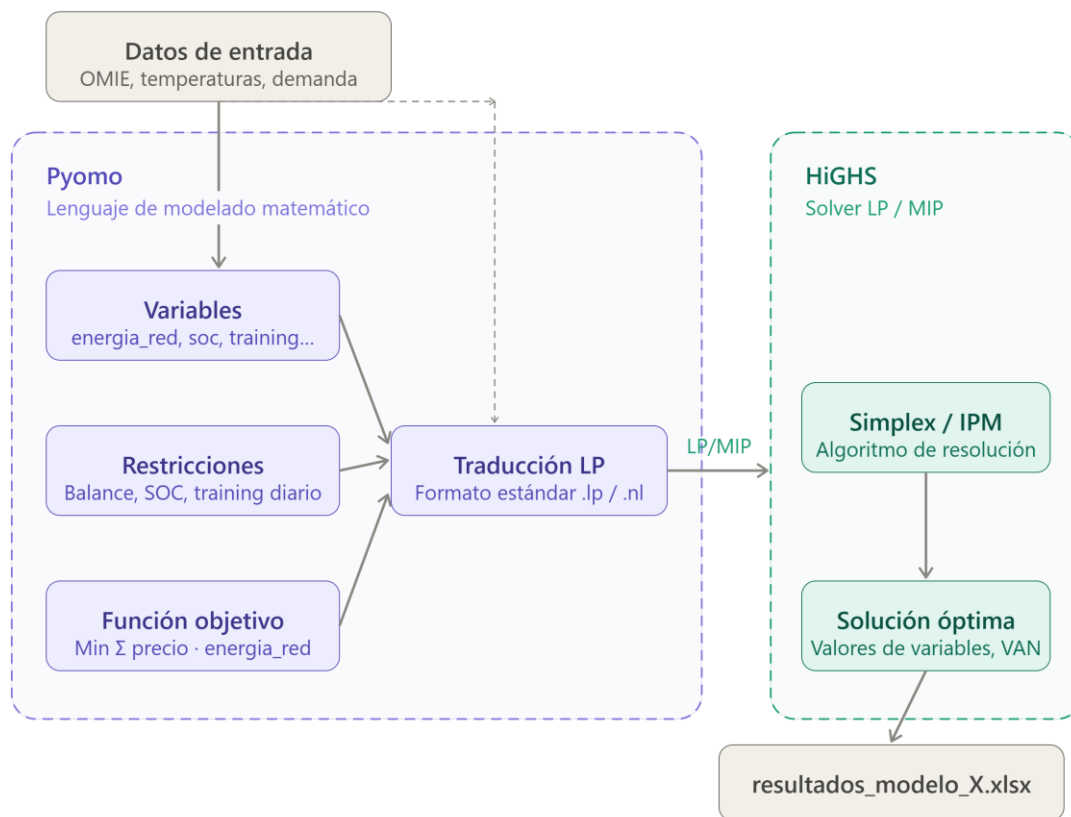


Figura 6. Diagrama de bloques del funcionamiento de Pyomo y HiGHS

Capítulo 3. ESTADO DE LA CUESTIÓN

3.1 EVOLUCIÓN DE LA GESTIÓN ENERGÉTICA EN CENTROS DE DATOS

La eficiencia energética en centros de datos es una línea de trabajo consolidada tanto en la literatura académica como en la industria. Los primeros enfoques se centraban en reducir el consumo global de la instalación mediante mejoras en el PUE, la refrigeración, la distribución eléctrica y la gestión térmica de los equipos [11], [2].

Sin embargo, el crecimiento reciente de la inteligencia artificial ha ampliado el problema. En centros de datos dedicados a IA, la elevada potencia instalada, la concentración de cargas intensivas en GPU y la operación continua hacen que el coste energético dependa no solo de cuánta energía se consume, sino también de cuándo se consume [1].

Por ello, la literatura más reciente empieza a tratar el centro de datos como un consumidor flexible, capaz de adaptar parte de su operación a señales externas, como precios eléctricos, disponibilidad renovable o intensidad de carbono. Este cambio de enfoque es relevante para el presente trabajo, ya que permite pasar de una visión estática del consumo energético a una formulación optimizable, donde la operación horaria del centro de datos se convierte en una variable de decisión.

3.2 DEMAND RESPONSE Y FLEXIBILIDAD DE CARGA EN CPDS DE IA

El demand response, o respuesta de la demanda, consiste en modificar el perfil de consumo eléctrico en respuesta a señales de precio, incentivos económicos o necesidades del sistema eléctrico. En centros de datos, esta idea resulta especialmente interesante porque, como ya

se ha mencionado en capítulos anteriores, determinadas cargas computacionales pueden desplazarse dentro de ciertos límites temporales.

Los primeros trabajos sobre gestión flexible de la demanda en sistemas informáticos de gran escala ya planteaban la posibilidad de reducir la factura eléctrica desplazando carga entre momentos o ubicaciones con distinto precio de la energía [12]. Más recientemente, la literatura sobre carbon-aware computing ha ampliado este enfoque, analizando cómo los centros de datos pueden adaptar su operación a señales como la intensidad de carbono o la disponibilidad de generación renovable [13].

En los centros de datos dedicados a IA, la distinción entre entrenamiento e inferencia permite trasladar esta idea al modelo energético. La inferencia debe atenderse prácticamente en tiempo real, mientras que parte del entrenamiento puede admitir cierto desplazamiento temporal, siempre que se complete dentro del horizonte previsto. Esta flexibilidad permite desplazar parte del consumo hacia horas de menor precio eléctrico, aprovechando la variabilidad de los precios horarios del mercado [1], [18].

No obstante, muchos enfoques se centran en la gestión de la carga IT sin integrar de forma conjunta otros elementos relevantes para un centro de datos real, como la refrigeración dependiente de la temperatura exterior, el almacenamiento energético o el análisis económico de la inversión. Este trabajo parte de esa flexibilidad computacional, pero la combina con otras palancas operativas para evaluar su impacto conjunto.

3.3 ALMACENAMIENTO EN BATERÍA Y ARBITRAJE ENERGÉTICO

El almacenamiento en batería se ha consolidado como una herramienta relevante para consumidores de gran escala, especialmente por la caída de costes de la tecnología Li-ion y por el aumento de la volatilidad de los mercados eléctricos [7]. En un contexto de precios horarios variables, un BESS permite desplazar energía en el tiempo: cargarse en periodos de bajo precio y descargarse cuando el precio es mayor. Este uso, conocido como arbitraje

energético, es una de las aplicaciones más directas del almacenamiento en mercados eléctricos horarios.

La literatura económica muestra, sin embargo, que la rentabilidad de un BESS depende de muchas variables: el coste de inversión, la eficiencia de carga y descarga, la degradación, la vida útil, el perfil de precios y la estrategia de operación [8]. Por ello, los estudios de viabilidad suelen requerir análisis de sensibilidad, especialmente cuando el diferencial de precios capturable no es constante a lo largo del tiempo.

En centros de datos, el almacenamiento presenta una característica adicional: estas instalaciones ya incorporan sistemas de respaldo eléctrico para garantizar la continuidad del servicio. Esto diferencia al CPD de otros consumidores industriales, porque parte de la infraestructura asociada a la seguridad de suministro puede solaparse con el uso de baterías. La literatura técnica ha señalado el interés de las baterías Li-ion como alternativa o complemento a sistemas SAI tradicionales, especialmente en aplicaciones donde la densidad energética, el mantenimiento y la respuesta dinámica son relevantes [16].

Aun así, el uso de un BESS en centros de datos no debe analizarse únicamente como una instalación de arbitraje puro. En este tipo de infraestructuras, una parte de los elementos comunes de gestión, conexión, seguridad, refrigeración y respaldo ya viene exigida por la propia necesidad de disponibilidad del centro de datos. Por ello, el interés económico del BESS depende de si el sobredimensionamiento o aprovechamiento adicional de esa infraestructura permite capturar ahorros horarios suficientes para compensar el CAPEX incremental y la pérdida progresiva de capacidad.

Por ello, en este TFG el BESS se incorpora no solo como una restricción operativa, sino también como una inversión cuya viabilidad debe evaluarse económicamente.

3.4 REFRIGERACIÓN, CLIMA Y ALMACENAMIENTO TÉRMICO

La refrigeración aparece de forma recurrente en la literatura sobre eficiencia energética en centros de datos, especialmente por su impacto sobre el PUE y sobre el consumo auxiliar de la instalación [2]. En los centros de datos de IA, este aspecto gana importancia por la elevada densidad de potencia de los equipos y por la necesidad de evacuar grandes cargas térmicas de forma continua [5].

Muchos modelos simplificados representan la refrigeración mediante un PUE constante. Esta aproximación es útil para estimaciones generales, pero no permite captar el efecto horario de la temperatura exterior sobre el rendimiento del sistema frigorífico. Si el consumo de refrigeración cambia con la temperatura, entonces el coste eléctrico horario del centro de datos no depende solo de la carga IT y del precio de mercado, sino también de las condiciones climáticas.

El rendimiento de los sistemas de compresión depende de las temperaturas de operación y del ambiente exterior [6]. Esto justifica incorporar una representación climática, aunque sea simplificada, en modelos económicos de operación. En este trabajo no se simula con detalle todo el sistema térmico, pero sí se introduce una demanda de refrigeración dependiente de la temperatura exterior, construida a partir de datos meteorológicos horarios [19].

Además, el almacenamiento térmico de frío, o CTES, permite desplazar en el tiempo la producción de frío. Frente a una batería eléctrica, que almacena energía eléctrica para cualquier uso posterior, el CTES actúa directamente sobre la demanda de refrigeración. Esta diferencia lo convierte en una alternativa especialmente interesante para centros de datos, porque vincula el almacenamiento con la temperatura ambiente, el rendimiento del sistema frigorífico y la carga térmica de la instalación. En la literatura, este tipo de almacenamiento aparece más desarrollado en edificios e instalaciones industriales que en centros de datos dedicados a IA, por lo que su incorporación constituye una de las extensiones diferenciales de este trabajo.

3.5 ANÁLISIS ECONÓMICO DE INVERSIONES ENERGÉTICAS

La evaluación de medidas de eficiencia o flexibilidad energética requiere combinar el modelo técnico de operación con herramientas económicas. Métricas como el valor actual neto, la tasa interna de rentabilidad o el periodo de retorno permiten valorar si una inversión compensa los ahorros futuros que genera [17]. En el caso de centros de datos, esta evaluación es especialmente relevante porque las medidas analizadas suelen requerir inversiones elevadas y se justifican por ahorros acumulados durante varios años.

La incertidumbre del precio eléctrico es uno de los factores más importantes. Una estrategia basada en desplazar consumo o arbitrar con almacenamiento será más atractiva cuanto mayor sea la diferencia entre horas baratas y caras. Sin embargo, esa diferencia puede variar según el año, la penetración renovable, la demanda del sistema o la regulación del mercado. Por ello, los análisis deterministas suelen complementarse con análisis de sensibilidad o escenarios alternativos, que permiten comprobar cómo cambia la rentabilidad ante variaciones en precios, CAPEX, eficiencia o vida útil [15].

En el caso del almacenamiento, también resulta útil comparar alternativas mediante métricas de coste nivelado, como LCOE o LCOS, especialmente cuando se analizan activos con distinta vida útil, inversión inicial y perfil de operación [14]. En este trabajo, estas herramientas se emplean como complemento al análisis operativo, con el objetivo de valorar no solo qué estrategia reduce más el coste eléctrico anual, sino también en qué condiciones económicas merece la pena invertir en los activos que permiten capturar ese ahorro.

3.6 HUECO IDENTIFICADO Y APORTACIÓN DE ESTE TFG

La revisión anterior muestra que existen líneas de trabajo consolidadas sobre eficiencia energética en centros de datos, flexibilidad de cargas computacionales, almacenamiento en batería, refrigeración industrial y análisis económico de inversiones. Sin embargo, estos elementos suelen estudiarse por separado o con enfoques parciales. Algunos trabajos se

centran en la optimización de cargas, otros en la reducción del PUE, otros en el arbitraje con baterías y otros en el análisis económico de activos energéticos.

Este TFG se sitúa en la intersección de esas líneas. Su aportación no consiste en desarrollar una tecnología nueva, sino en integrar varias palancas de gestión energética dentro de un mismo marco de optimización aplicado a un centro de datos de IA de gran escala conectado al mercado eléctrico ibérico. La metodología incremental permite partir de un caso base sin gestión activa e incorporar progresivamente flexibilidad de entrenamiento, refrigeración dependiente de la temperatura, almacenamiento mediante BESS, almacenamiento térmico de frío y, finalmente, la combinación de ambos almacenamientos en un sistema híbrido.

Esta estructura permite medir el impacto económico de cada estrategia por separado y también su efecto conjunto. De este modo, el trabajo aporta un marco simplificado, reproducible y adaptable, orientado a responder una pregunta práctica: hasta qué punto puede reducirse el coste eléctrico anual de un centro de datos de IA mediante gestión activa de la demanda y en qué condiciones económicas resulta justificable invertir en los activos necesarios para hacerlo posible.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

La revisión del estado de la cuestión muestra que la gestión energética activa en centros de datos dedicados a inteligencia artificial es un ámbito de interés creciente, tanto por el aumento del consumo eléctrico del sector como por la volatilidad de los precios horarios del mercado eléctrico. Aunque existen trabajos sobre eficiencia energética, flexibilidad de carga, refrigeración y almacenamiento, estos elementos suelen analizarse de forma separada. Por ello, resulta útil plantear un marco integrado que permita estudiar cómo interactúan entre sí y qué impacto económico tiene cada uno.

Desde el punto de vista técnico, un CPD de IA combina cargas con comportamientos diferentes. La inferencia debe atenderse prácticamente en tiempo real, mientras que parte del entrenamiento puede desplazarse dentro de ciertos límites temporales. A esta flexibilidad se suma el consumo de refrigeración, condicionado por la temperatura exterior, y la posibilidad de incorporar almacenamiento eléctrico o térmico. La combinación de estos elementos da lugar a un problema de optimización horario en el que las decisiones de operación no son independientes: desplazar carga, cargar una batería o producir frío en otro momento puede modificar el coste total de forma conjunta.

Desde el punto de vista económico, las estrategias de gestión activa solo tienen sentido si el ahorro operativo obtenido compensa la inversión necesaria para aplicarlas. En particular, los sistemas de almacenamiento requieren un desembolso inicial elevado y su rentabilidad depende de variables inciertas, como el precio eléctrico, el coste de la tecnología, la eficiencia y la vida útil del activo. Por ello, no basta con calcular un ahorro anual en un escenario central, sino que es necesario analizar la sensibilidad de los resultados ante cambios en los principales parámetros.

El trabajo se justifica, por tanto, por la necesidad de cuantificar de forma ordenada el valor económico de distintas estrategias de gestión energética en un CPD de IA. Para ello se plantea una metodología incremental, en la que se parte de un caso base y se incorporan progresivamente flexibilidad de carga, refrigeración dependiente de la temperatura, almacenamiento mediante BESS y almacenamiento térmico de frío. Esta estructura permite medir no solo el ahorro total alcanzable, sino también la contribución de cada estrategia al resultado final.

4.2 OBJETIVOS

El objetivo principal de este Trabajo Fin de Grado es desarrollar y comparar distintos modelos de optimización energética aplicados a un centro de datos dedicado a inteligencia artificial, con el fin de cuantificar el ahorro económico que puede obtenerse mediante gestión activa de la demanda, refrigeración dependiente de la temperatura exterior y sistemas de almacenamiento eléctrico y térmico.

De este objetivo general se derivan los siguientes objetivos específicos:

- O1. Definir un caso base representativo de un centro de datos de IA de gran escala, con una potencia IT instalada de 20 MW y un reparto entre cargas rígidas y flexibles.
- O2. Plantear y resolver modelos de optimización con un nivel de complejidad creciente, incorporando progresivamente la flexibilidad temporal de la carga de entrenamiento, la refrigeración dependiente de la temperatura exterior, el almacenamiento eléctrico mediante BESS, el almacenamiento térmico de frío mediante CTES y la combinación de ambos en un sistema híbrido.
- O3. Cuantificar el ahorro en coste energético de cada modelo respecto al caso base, identificando la contribución individual de cada estrategia al ahorro total.

- O4. Evaluar la viabilidad económica de los modelos que requieren inversión en infraestructura, calculando el VAN, la TIR y el periodo de retorno.
- O5. Analizar la robustez de los resultados frente a la incertidumbre en los parámetros clave, mediante un análisis de sensibilidad multivariable.
- O6. Comparar el comportamiento económico y operativo del almacenamiento eléctrico mediante BESS frente al almacenamiento térmico de frío mediante CTES, así como su combinación en un sistema híbrido, valorando en qué condiciones cada alternativa resulta más conveniente y si la inversión marginal en cada tecnología se justifica una vez instalada la otra.

4.3 METODOLOGÍA

La metodología se estructura en cuatro fases sucesivas.

Fase 1. Definición del caso base y preparación de datos.

Se fijan las características técnicas del centro de datos (potencia IT, PUE, reparto de cargas) y el horizonte del análisis, un año completo de 8.760 horas. A continuación, se obtienen y procesan los datos de precio horario del mercado OMIE y se construye el perfil de demanda horaria del CPD, separando las componentes de carga IT, refrigeración y consumos auxiliares.

Fase 2. Desarrollo de los modelos de optimización.

Se formulan siete modelos de optimización de complejidad creciente, implementados en Python mediante Pyomo y resueltos con HiGHS. Cada modelo incorpora nuevas variables, parámetros y restricciones respecto al anterior, siguiendo una construcción incremental. Esta estructura permite aislar el efecto de cada palanca de gestión energética: flexibilidad de entrenamiento, refrigeración dependiente de la temperatura, almacenamiento mediante BESS y almacenamiento térmico de frío mediante CTES.

Fase 3. Análisis económico de la inversión.

A partir de los ahorros en coste energético de cada modelo se construyen los flujos de caja del proyecto bajo distintos escenarios de crecimiento del precio eléctrico. Sobre esos flujos se calculan los indicadores de rentabilidad habituales (VAN, TIR, payback) y se realiza un análisis de sensibilidad bidimensional sobre los parámetros con mayor incertidumbre.

Fase 4. Comparación de resultados y análisis de sensibilidad.

Finalmente, se comparan los resultados de todos los modelos respecto al caso base, identificando el ahorro anual, la contribución de cada estrategia y las condiciones bajo las que cada alternativa resulta más conveniente. El análisis se completa con estudios de sensibilidad sobre los parámetros con mayor incertidumbre, como el precio eléctrico, el CAPEX, la eficiencia del sistema y la vida útil de los activos. Esta última fase permite valorar la robustez de las conclusiones y detectar qué variables condicionan en mayor medida la viabilidad económica del proyecto.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

El desarrollo del proyecto se ha organizado en las siguientes fases, que se han desarrollado entre finales de enero y julio de 2026, compaginándose con los periodos de exámenes del curso:

Fase 1, análisis preliminar y definición del caso base:

Revisión bibliográfica, definición de la arquitectura del CPD modelado, obtención y preprocesado de los datos de precio y demanda, y validación del perfil horario anual.

Fase 2, desarrollo de los Modelos 1 a 3:

Implementación del caso base (Modelo 1), del modelo con flexibilidad anual del training (Modelo 2) y del de flexibilidad diaria (Modelo 3), con validación cruzada de los resultados y análisis de coherencia entre ellos.

Fase 3, desarrollo de los Modelos 4 a 7:

Incorporación del modelo climático de refrigeración (Modelo 4), del sistema de almacenamiento en batería (Modelo 5) y del sistema de almacenamiento térmico en frío CTES (Modelo 6) y del sistema híbrido que combina ambos almacenamientos (Modelo 7), quedando así montados todos los modelos de optimización del proyecto.

Fase 4, análisis económico y de riesgo:

Construcción del modelo de flujos de caja para los siete modelos ya implementados, cálculo de indicadores de rentabilidad (TIR, VAN, payback) y análisis de sensibilidad.

Fase 5, redacción de la memoria:

Documentación de cada modelo, análisis detallado de los resultados y redacción de las conclusiones, con una breve interrupción puntual por exámenes de junio.

Fase 6, afinar:

Revisión de la memoria con el tutor (JPC), incorporación de sus comentarios y pulido final de resultados, gráficas y coherencia del documento.

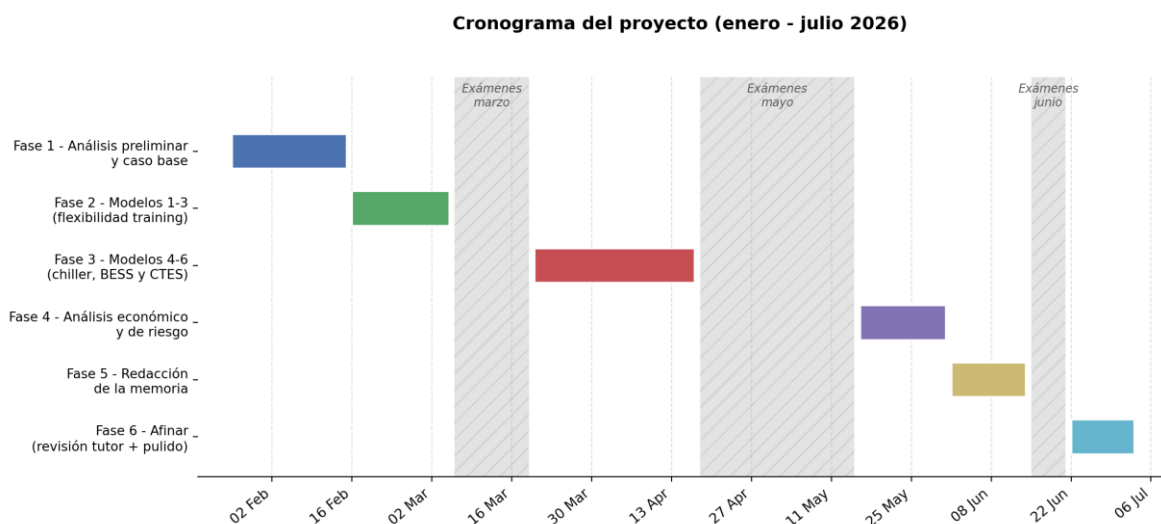


Figura 7. GANTT del proyecto

En cuanto a la estimación económica, al tratarse de un trabajo académico de investigación y desarrollo, los costes se reducen al tiempo dedicado por la autora del documento y al uso de software de código abierto (Python, Pyomo, HiGHS, pandas), sin coste de licencia. El

recurso principal consumido es el coste de oportunidad del tiempo invertido, estimado en un rango de entre 200 y 250 horas a lo largo del proyecto.

4.5 ALINEACIÓN CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE (ODS)

Más allá de su vertiente económica, el trabajo conecta con varios Objetivos de Desarrollo Sostenible (ODS) de Naciones Unidas, en concreto los ODS 7, 9, 12 y 13. Las estrategias planteadas, gestión activa de la demanda y almacenamiento de energía, inciden directamente en la eficiencia y la sostenibilidad del sistema eléctrico y se alinean con la transición energética en marcha.

ODS 7. Energía asequible y no contaminante.

El trabajo encaja con el ODS 7 al empujar una transición hacia una energía más limpia y asequible: optimiza el uso de los recursos disponibles y rebaja los picos de demanda del centro de datos. El desplazamiento del entrenamiento hacia las horas valle del mercado coincide habitualmente con tramos de mayor penetración renovable, así que de forma indirecta también se reduce la huella de carbono del consumo, y todo ello sin tener que invertir en nuevos activos de generación.

ODS 9. Industria, innovación e infraestructura.

Encaja también con el ODS 9, porque introduce innovación tecnológica en la gestión energética de una infraestructura digital crítica. El modelo integrado del centro de datos, que combina flexibilidad de carga, refrigeración calibrada con la temperatura exterior y almacenamiento en batería, funciona como herramienta de apoyo a la decisión y empuja la eficiencia y la modernización como ejes de la operación. La metodología, además, es replicable: con pocas modificaciones se puede aplicar a otros emplazamientos, lo que ayuda a difundir buenas prácticas en el sector.

ODS 12. Producción y consumo responsables.

En el caso del ODS 12, el trabajo apoya una gestión responsable de los recursos energéticos al poner cifras concretas al ahorro que pueden generar las estrategias operativas y de inversión orientadas a un consumo más eficiente. Los ahorros obtenidos no implican necesariamente una reducción equivalente de la energía total consumida, sino una reducción del coste de adquisición al desplazar parte de la demanda hacia horas de menor precio. En la medida en que esas horas coincidan con mayor penetración renovable, la estrategia puede contribuir indirectamente a reducir la intensidad de carbono del consumo.

ODS 13. Acción por el clima.

El trabajo contribuye también al ODS 13 al reducir de forma sistemática las emisiones asociadas a la operación del centro de datos y al integrar indirectamente energía renovable a través del desplazamiento temporal de la carga. Hay además un detalle interesante: como los resultados son muy sensibles a la volatilidad del mercado, y como la entrada masiva de renovables tiende justamente a aumentar esa volatilidad, el valor económico de la flexibilidad y de la batería irá probablemente al alza a medida que avance la descarbonización, reforzando el argumento para su despliegue.

El proyecto suma así por varios frentes hacia un desarrollo tecnológico más sostenible, y muestra que rentabilidad económica y eficiencia energética no son objetivos contrapuestos: pueden ir de la mano.

Capítulo 5. SISTEMA/MODELO DESARROLLADO

5.1 DESCRIPCIÓN DEL CASO BASE: EL CENTRO DE DATOS MODELADO

El sistema objeto de estudio es un centro de datos de gran escala dedicado a cargas mixtas de inteligencia artificial. Sus características técnicas se recogen en la Tabla 2.

Tabla 2. Parámetros técnicos del centro de datos modelado.

Parámetro	Valor
Potencia IT instalada	20 MW
PUE	1,3
Potencia total demandada	26 MW
Carga de entrenamiento (flexible)	60% IT → 12 MW
Carga de inferencia (rígida)	40% IT → 8 MW
Energía anual estimada	~228.000 MWh
Horizonte de optimización	8.760 horas (año completo)

La demanda horaria total del CPD se descompone en cuatro componentes: la carga de entrenamiento, la de inferencia, el consumo de refrigeración y los consumos auxiliares (iluminación, distribución eléctrica, sistemas de control). Esta separación es clave para los modelos de optimización, porque solo la carga de entrenamiento admite desplazamiento temporal.

5.2 DATOS DE ENTRADA: PRECIOS OMIE Y PERFIL DE DEMANDA

Los datos de entrada del modelo corresponden al año 2025 y comprenden 8.760 registros horarios con las siguientes variables:

Precios eléctricos: se utilizan los precios horarios del mercado spot ibérico (OMIE) para el año 2025 [18], expresados en €/MWh. El precio medio anual es de 65,28 €/MWh, con una desviación típica de 47,62 €/MWh, un máximo de 240,00 €/MWh y un mínimo de -15,00 €/MWh. Esta elevada variabilidad es la que hace económicamente atractivo el desplazamiento de cargas hacia horas de precio reducido.

Temperatura exterior: datos horarios de temperatura exterior registrados en Alcobendas (Madrid) durante 2025, obtenidos a través de la API de Open-Meteo [19]. Estos datos se utilizan en los Modelos 4 y 5 para calibrar el consumo horario del sistema de refrigeración en función de las condiciones ambientales.

Perfil de demanda: el perfil horario de cada componente de la carga (training, inferencia, climatización, comunes) se construye a partir de los parámetros técnicos del CPD, distribuyendo la potencia total según las proporciones definidas en el caso base y aplicando el PUE correspondiente. Cabe señalar que este perfil corresponde a un centro de datos teórico y ha sido generado mediante inteligencia artificial, reproduciendo el comportamiento esperable de un CPD real de estas características, ya que no ha sido posible acceder a datos reales de operación de un centro de datos por motivos de confidencialidad de las empresas del sector.

5.3 MODELO 1: CASO BASE SIN OPTIMIZACIÓN

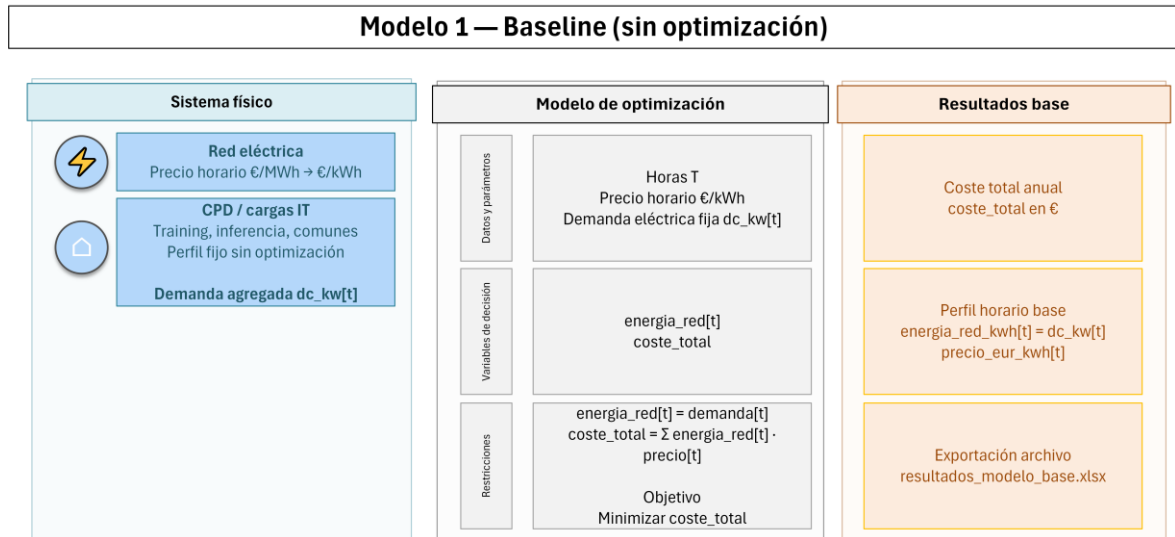


Figura 8 – Diagrama base Modelo 1

5.3.1 DESCRIPCIÓN

El Modelo 1 es el punto de referencia del análisis. Representa la operación de un CPD sin gestión activa de la demanda: el centro compra de la red, en cada hora, exactamente la energía que necesita sea cual sea el precio. No hay ningún grado de libertad en la decisión de compra.

Cumple una función doble. Fija el coste de referencia con el que se miden los ahorros de los modelos posteriores y, a la vez, verifica la coherencia de los datos de entrada, ya que el coste calculado debe coincidir con la simple multiplicación de la demanda horaria por el precio horario.

5.3.2 FORMULACIÓN MATEMÁTICA

Sea $T = \{1, 2, \dots, 8760\}$ el conjunto de horas del año. Para cada hora $t \in T$ se definen los siguientes parámetros:

pt : precio de la energía en la hora t [€/kWh]

dt : demanda total del CPD en la hora t [kW]

La única variable de decisión es:

et^{red} : energía comprada de la red en la hora t [kWh]

Restricción de balance:

$$et^{red} = dt \quad \forall t \in T$$

Ecuación 4 – Energía adquirida de red en periodo horario

Función objetivo:

$$\text{mín } \sum_{t \in T} et^{red} \cdot pt$$

Ecuación 5 – Función objetivo utilizada

Al no existir flexibilidad alguna, la solución óptima es trivialmente $etred = dt$ para todo t , y el coste resultante es la suma del producto demanda-precio hora a hora. El coste anual del Modelo 1 asciende a 8.903.875 €, que constituye el coste de referencia del análisis.

5.4 MODELO 2: FLEXIBILIDAD DE TRAINING ANUAL

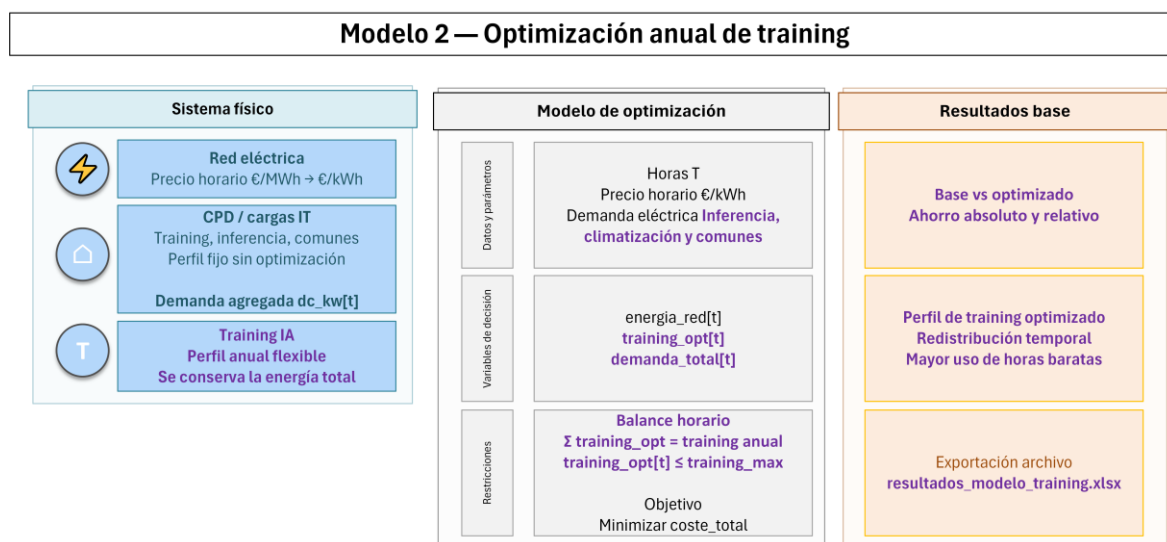


Figura 9 – Diagrama base Modelo 2

5.4.1 DESCRIPCIÓN Y MOTIVACIÓN

El Modelo 2 introduce el primer grado de libertad: la carga de entrenamiento puede repartirse libremente a lo largo del año, con una única condición, que la energía total de training del año coincida con la del caso base. No se impone nada sobre los días ni las horas concretas en que se ejecuta.

Este modelo marca el límite teórico superior del ahorro por desplazamiento de carga, porque da al optimizador toda la flexibilidad imaginable. Concentrar meses de entrenamiento en unas pocas horas baratas no es viable en la práctica, pero el resultado sirve como cota superior de referencia.

5.4.2 FORMULACIÓN MATEMÁTICA

Se añade la variable:

x_t : energía de training asignada a la hora t [kWh]

La demanda total en cada hora es:

$$dt = xt + inft + climt + comt \quad \forall t \in T$$

Ecuación 6 – Demanda total horaria

donde *inft*, *climt* y *comt* son los consumos horarios de inferencia, climatización y comunes respectivamente, todos ellos fijos.

Restricciones:

$$\sum_{t \in T} xt = E_{training\ annual} \quad (\text{energía anual de training conservada})$$

Ecuación 7 – Energía anual de training conservada

$$0 \leq xt \leq x_{m\acute{a}x} \quad \forall t \in T \quad (\text{límite de potencia máxima})$$

Ecuación 8 – Límite de potencia máxima

donde $x_{m\acute{a}x}$ es la potencia máxima histórica de training observada en el dataset.

Función objetivo:

$$\text{mín} \sum_{t \in T} (xt + inft + climt + comt) \cdot pt$$

Ecuación 9 – Función objetivo actualizada

El optimizador concentra todo el training posible en las horas más baratas del año. El coste anual del Modelo 2 es de 7.358.840 €, un ahorro de 1.545.035 € (17,35%) frente al caso base.

5.5 MODELO 3: FLEXIBILIDAD DE TRAINING DIARIA

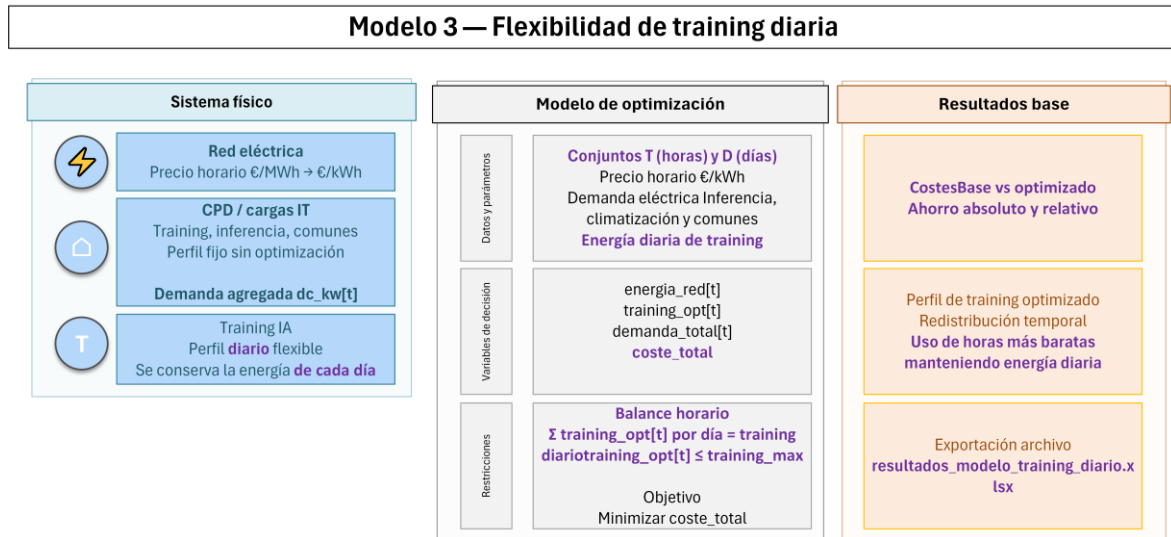


Figura 10 – Diagrama base Modelo 3

5.5.1 DESCRIPCIÓN Y MOTIVACIÓN

El Modelo 3 añade una restricción operativa clave respecto al Modelo 2: la energía de training de cada día debe ejecutarse íntegramente dentro de ese mismo día. No cabe trasladar training de un día a otro; solo se redistribuye entre las 24 horas de la jornada.

Esta condición refleja mucho mejor la realidad de un CPD de IA, donde los trabajos de entrenamiento tienen plazos acotados y los equipos de ingeniería necesitan resultados en un horizonte razonable. La flexibilidad diaria es la forma más común de demand response en entornos industriales.

5.5.2 FORMULACIÓN MATEMÁTICA

Sea D el conjunto de días del año y $Td \subset T$ el subconjunto de horas pertenecientes al día d . La restricción de energía diaria sustituye a la restricción anual del Modelo 2:

$$\sum_{t \in T_d} x_t = E_{\text{training},d} \quad \forall d \in D$$

Ecuación 10 – Restricción de energía diaria

donde $E_{\text{training},d}$ es la energía de training que debe completarse el día d .

El resto de la formulación es idéntico al Modelo 2.

5.5.3 RESULTADOS

Al recortar mucho el espacio de redistribuciones posibles, el ahorro cae también de forma apreciable. El coste anual del Modelo 3 es de 8.080.644 €, con un ahorro de 823.231 € (9,25%) respecto al caso base. Los 722.000 €/año de diferencia con el Modelo 2 cuantifican el valor de la flexibilidad inter-diaria, que en la práctica no resulta aprovechable.

La Figura 10 revela además un problema operativo del Modelo 2, ya que sin restricción de agrupación diaria, el entrenamiento se concentra en los días más baratos del año, llegando a encadenar 11 días consecutivos (13 a 23 de enero) con menos del 5% de la energía media diaria. Un centro de datos real no admite semanas casi sin entrenar, además de que la desviación típica diaria del Modelo 2 (83.048 kWh) más que duplica la del Modelo 3 (37.055 kWh).

El origen del problema es la información disponible. Al Modelo 3 le basta con los precios del día en curso, que el mercado diario publica con antelación, mientras que el Modelo 2 necesitaría conocer con certeza los precios de todo el año. Por eso la recomendación operativa de este trabajo es el Modelo 3 y no el Modelo 2.

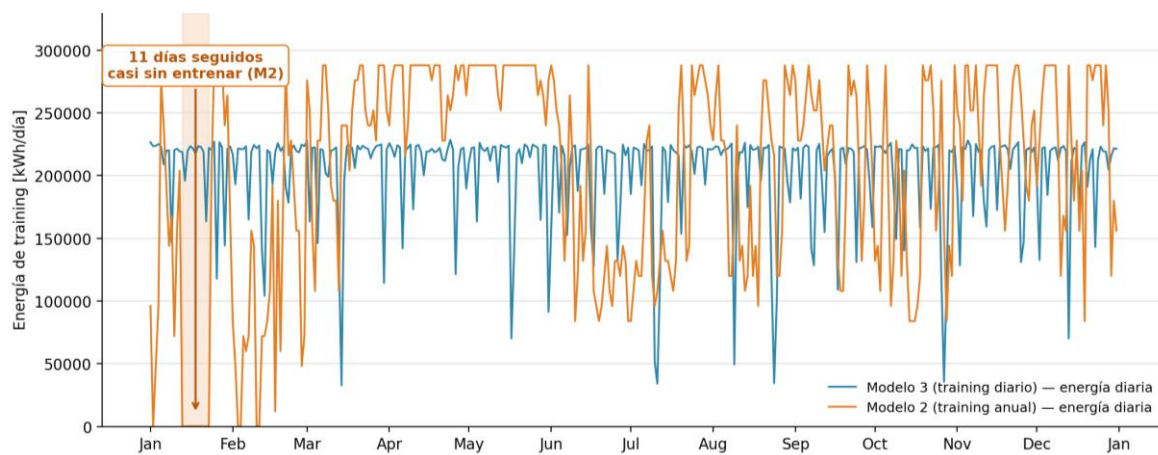


Figura 11. Energía de training diaria Modelo 2 frente a Modelo 3.

5.6 MODELO 4: MODELO CLIMÁTICO Y REFRIGERACIÓN POR AMONIACO

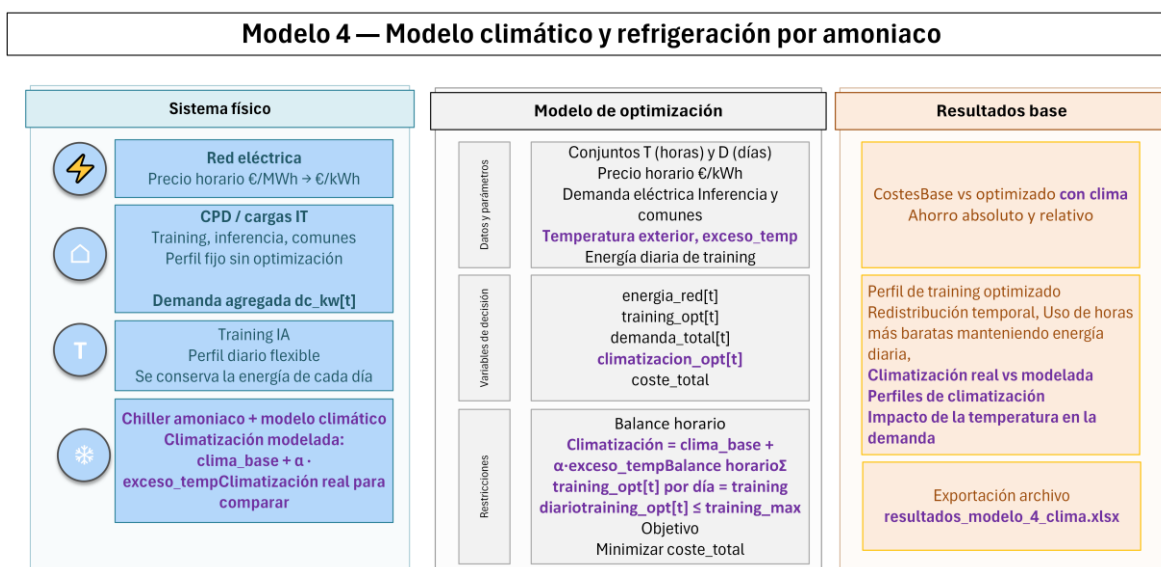


Figura 12 – Diagrama base Modelo 4

5.6.1 DESCRIPCIÓN Y MOTIVACIÓN

El Modelo 4 incorpora una representación explícita del sistema de refrigeración. En los modelos anteriores, el consumo de climatización se tomaba del dataset como un valor fijo hora a hora. Aquí ese consumo se sustituye por un modelo calibrado que estima el consumo eléctrico de refrigeración según la temperatura exterior.

La motivación es doble. Por un lado, permite representar de forma más realista la estacionalidad del consumo de refrigeración. Por otro, sienta las bases para el Modelo 6, en el que se incorpora almacenamiento térmico de frío y, por tanto, resulta necesario disponer de una variable asociada a la demanda de refrigeración.

5.6.2 MODELO DE DEMANDA DE REFRIGERACIÓN

El consumo eléctrico de refrigeración del CPD se modela como una función lineal de la temperatura exterior:

$$e_t^{frío} = e_{frío}^{base} + \alpha \cdot \max(T_t^{ext} - T^{ref}, 0)$$

Ecuación 11 – Consumo eléctrico de refrigeración dependiente de la temperatura

donde:

$e_t^{frío}$ es el consumo eléctrico de refrigeración en la hora (t) [kWh]

$e_{frío}^{base}$ es el consumo eléctrico base de refrigeración, estimado como el mínimo histórico del consumo de climatización [kWh].

$T^{ref} = 20$ °C es la temperatura de referencia por encima de la cual se activa el enfriamiento adicional

α es el coeficiente de sensibilidad climática [kWh/°C], calibrado a partir de los datos históricos

$T_{t^{ext}}$ es la temperatura exterior en la hora t [°C]

En este modelo, la refrigeración se representa directamente en unidades eléctricas equivalentes. Por tanto, no se introduce todavía una conversión térmica explícita mediante COP. Esta simplificación permite mantener la coherencia con el perfil histórico de climatización utilizado en los modelos anteriores y evita introducir parámetros físicos adicionales en una etapa cuyo objetivo principal es calibrar la dependencia climática del consumo de refrigeración.

La demanda total en cada hora incorpora ahora el consumo de refrigeración modelado en lugar del consumo de climatización del dataset:

$$dt = xt + inft + comt + e_t^{frío} \quad \forall t \in T$$

Ecuación 12 – Demanda total horaria con sistema de amoniac

5.6.3 FUNCIÓN OBJETIVO

La función objetivo y el resto de las restricciones coinciden con el Modelo 3.

5.6.4 RESULTADOS

El coste anual del Modelo 4 es de 8.081.904 €, casi idéntico al del Modelo 3 (8.080.644 €). La diferencia, apenas 1.260 €, indica que sustituir el perfil histórico de climatización por el modelo climático calibrado no altera de forma significativa la estructura del problema. Esto valida la coherencia entre ambas representaciones y permite utilizar esta formulación como base para el modelo posterior con almacenamiento térmico. Su aportación es, por tanto, más técnica que económica.

5.7 MODELO 5: SISTEMA COMPLETO CON ALMACENAMIENTO EN BATERÍA

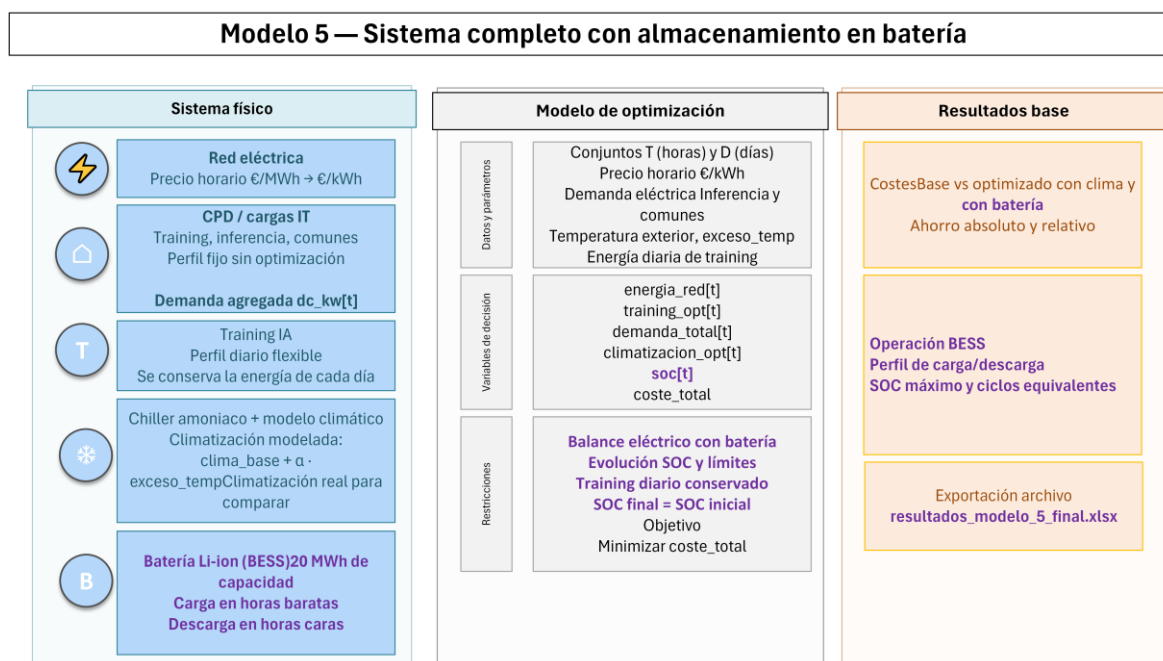


Figura 13 – Diagrama base Modelo 5

5.7.1 DESCRIPCIÓN Y MOTIVACIÓN

El Modelo 5 incorpora un sistema de almacenamiento eléctrico mediante batería sobre la base del Modelo 4, manteniendo la flexibilidad diaria de la carga de training y la refrigeración modelada en función de la temperatura exterior. El sistema considerado es un BESS de 20 MWh de capacidad nominal.

La batería desacopla en el tiempo la compra de energía a la red del consumo instantáneo del CPD: carga en horas de menor precio y descarga en horas de mayor precio, reduciendo así el coste anual de adquisición de electricidad. En este modelo, el BESS se utiliza como activo de gestión energética orientado al arbitraje horario sobre el mercado spot.

5.7.2 PARÁMETROS DE LA BATERÍA

Tabla 3. Parámetros del sistema BESS en el Modelo 5.

Parámetro	Valor
Capacidad nominal	20.000 kWh (20 MWh)
Potencia máxima de carga	5.000 kW
Potencia máxima de descarga	5.000 kW
Eficiencia de carga (η_c)	0,95
Eficiencia de descarga (η_d)	0,95
SOC inicial	10.000 kWh (50% de capacidad)

5.7.3 VARIABLES ADICIONALES

ct : energía cargada en la batería en la hora t [kWh]

st : energía descargada de la batería en la hora t [kWh]

$SOct$: estado de carga de la batería al final de la hora t [kWh]

z_t^{bat} : variable binaria de operación de la batería. Toma valor 1 cuando la batería puede cargar y valor 0 cuando puede descargar.

5.7.4 FORMULACIÓN MATEMÁTICA COMPLETA

Balance eléctrico: en cada hora, la energía procedente de la red más la descargada de la batería debe cubrir la demanda total más la energía destinada a cargar la batería:

$$et^{red} + st = dt + ct \quad \forall t \in T$$

Ecuación 13 – Balance eléctrico con batería

Evolución del SOC:

$$SOC_t = SOC_{t-1} + \eta_c \cdot ct - st / \eta_d \quad \forall t \in T$$

Ecuación 14 – Evolución del SOC de la batería

con $SOC_0 = 10.000$ kWh como condición inicial.

Límites del SOC:

$$0 \leq SOC_t \leq 20.000 \quad \forall t \in T$$

Ecuación 15 – Límites del SOC de la batería

Límites de potencia y no simultaneidad:

Para evitar que la batería cargue y descargue simultáneamente en una misma hora, se introduce una variable binaria de modo de operación. Esta variable toma valor 1 cuando la batería está habilitada para cargar y valor 0 cuando está habilitada para descargar:

$$0 \leq ct \leq 5.000 \cdot z_t^{bat} \quad \forall t \in T$$

Ecuación 16 – Límite de potencia de carga de la batería

$$0 \leq st \leq 5.000 \cdot (1 - z_t^{bat}) \quad \forall t \in T$$

Ecuación 17 – Límite de potencia de descarga de la batería

Con estas restricciones, en cada hora la batería puede operar en modo carga o en modo descarga, pero no en ambos simultáneamente. Por ello, el Modelo 5 se formula como un problema de programación lineal entera mixta (MILP), manteniendo el resto de la formulación sin cambios.

Condición cíclica anual: para garantizar que la batería no se «vacía» de forma artificial a lo largo del año, se impone que el SOC al final del último periodo sea igual al SOC inicial:

$$SOC_{8760} = SOC_0 = 10.000 \text{ kWh}$$

Ecuación 18 – Condición cíclica anual de la batería

Función objetivo:

$$\min \sum_{t \in T} et^{red} \cdot pt$$

Ecuación 19 – Función objetivo del Modelo 5

5.7.5 RESULTADOS

El coste anual del Modelo 5 es de 7.464.296 €, con un ahorro de 1.439.579 € (16,17 %) respecto al caso base. La batería trabaja en modo carga durante las horas de menor precio y en modo descarga durante las horas de mayor precio, respetando la restricción de no simultaneidad entre ambas operaciones. La diferencia entre la energía cargada y descargada recoge las pérdidas asociadas a la eficiencia del ciclo de carga y descarga. La batería no reduce la demanda eléctrica real del centro de datos, sino que desplaza su origen entre red y almacenamiento, por lo que no libera capacidad eléctrica adicional para carga IT. Como indicador del uso realizado del sistema, la batería gestiona 10.918.707 kWh/año en modo descarga (12.098.290 kWh/año en carga), equivalentes a aproximadamente 546 ciclos completos al año (Figura 12).

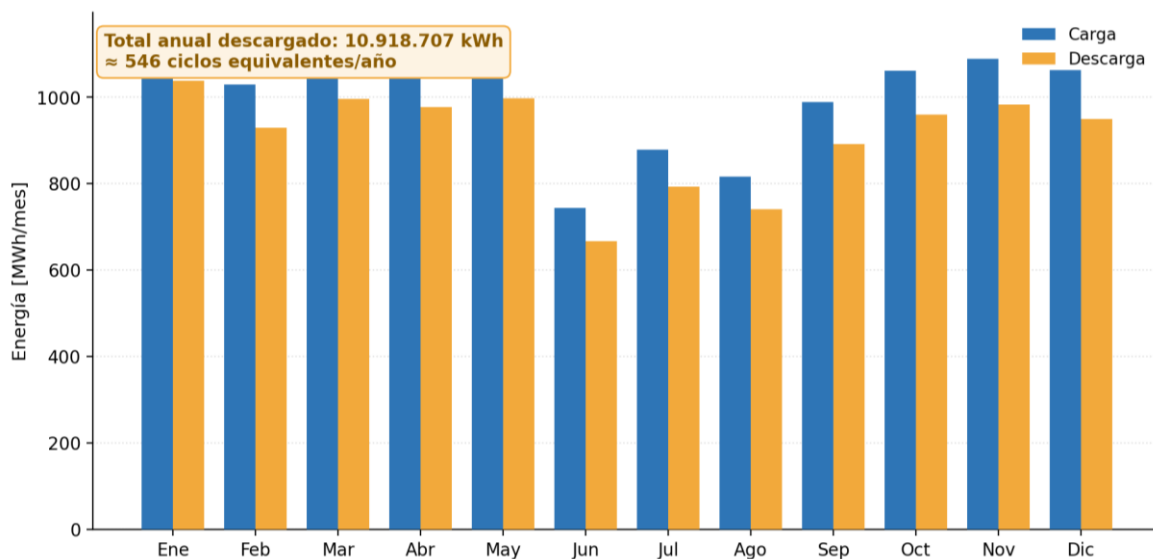


Figura 14. Energía gestionada por la batería BESS en el Modelo 5

5.8 MODELO 6: SISTEMA CTES CON ALMACENAMIENTO TÉRMICO DE FRÍO DE FRÍO

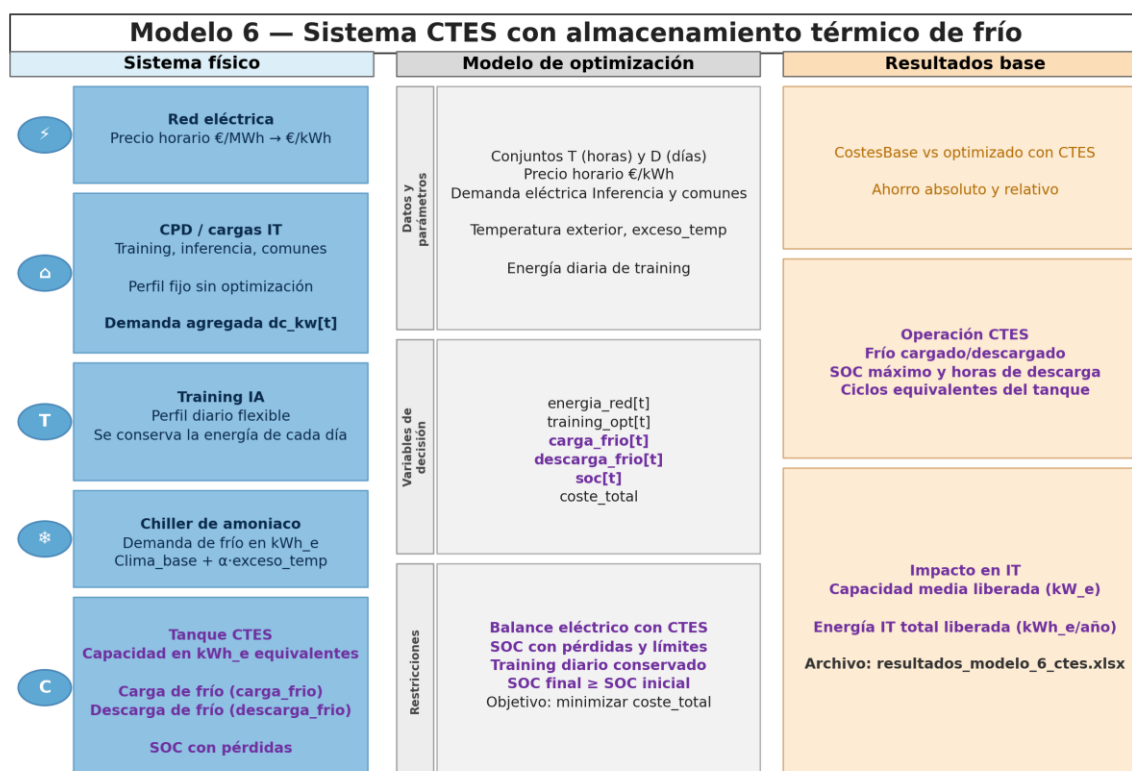


Figura 15 – Diagrama base Modelo 6

5.8.1 DESCRIPCIÓN Y MOTIVACIÓN

El Modelo 6 incorpora un sistema de almacenamiento térmico de frío, conocido como CTES (Cold Thermal Energy Storage), sobre la base del Modelo 4. No se plantea como una extensión acumulativa del Modelo 5, sino como una alternativa tecnológica al almacenamiento eléctrico mediante batería. Por ello, se construye sobre la misma base que el Modelo 5 (flexibilidad diaria de la carga de training y refrigeración modelada en función de la temperatura exterior), pero sustituye el BESS por un tanque de almacenamiento térmico.

El funcionamiento del sistema se basa en desplazar temporalmente la producción de frío. Durante las horas de menor precio eléctrico, el chiller de amoniaco produce frío adicional y lo almacena en el tanque. En las horas de mayor precio, el tanque descarga frío y reduce el consumo eléctrico necesario para refrigeración. De este modo, el CTES no actúa sobre toda la demanda eléctrica del CPD, como ocurre con el BESS, sino únicamente sobre la parte asociada a la refrigeración.

Esta diferencia permite comparar de forma homogénea dos alternativas de almacenamiento sobre una misma base operativa: el almacenamiento eléctrico, que modifica el balance eléctrico total del CPD, y el almacenamiento térmico, que desplaza específicamente la demanda de refrigeración. El interés del CTES se encuentra en que puede capturar parte del valor del arbitraje horario con un CAPEX inferior al sistema con BESS, 7,4 M€ frente a 11,0 M€, y sin necesidad de reemplazo dentro del horizonte de análisis de 15 años.

5.8.2 PARÁMETROS DEL SISTEMA

Los parámetros adoptados para el sistema CTES se recogen en la Tabla 4.

Tabla 4. Parámetros del sistema CTES en el Modelo 6.

Parámetro	Valor
Capacidad térmica nominal	30.000 kWh _{th} ($\approx$ 8.571 kWh _e con COP = 3,5)
Potencia máxima de carga	5.000 kWh _{th} /h ($\approx$ 1.429 kWh _e /h)
Potencia máxima de descarga	5.000 kWh _{th} /h ($\approx$ 1.429 kWh _e /h)
Pérdida térmica horaria	0,5 %
Eficiencia de carga (η_c)	0,95
Eficiencia de descarga (η_d)	0,95
SOC inicial	9.000 kWh _{th} (30 % de la capacidad), equivalente a 2.571 kWh _e
COP del chiller	3,5

El coste de inversión total del Modelo 6 asciende a 7.400.000 €, repartido entre el chiller de amoníaco (4.500.000 €), el tanque CTES (30.000 kWh_{th} $\times$ 80 €/kWh_{th} = 2.400.000 €) y los costes de integración y control (500.000 €). A diferencia del sistema BESS, el tanque de almacenamiento térmico no requiere reemplazo durante el horizonte de análisis de 15 años, ya que su vida útil supera ese plazo. El OPEX anual se estima en un 2 % del CAPEX del chiller (90.000 €/año) más un 1 % del CAPEX del tanque (24.000 €/año).

5.8.3 FORMULACIÓN MATEMÁTICA

El Modelo 6 introduce variables adicionales para describir el estado y la operación del tanque térmico. Aunque físicamente el almacenamiento se dimensiona en unidades térmicas [kWh_{th}], en la formulación se trabaja en unidades eléctricas equivalentes [kWh_e], obtenidas a partir del COP del chiller:

$$E_{eq} = Q_{th} / COP$$

Con $COP = 3,5$, la capacidad térmica de $30.000 \text{ kWh}_{\text{th}}$ equivale a 8.571 kWh_e . Del mismo modo, una potencia térmica máxima de carga o descarga de $5.000 \text{ kWh}_{\text{th}}/\text{h}$ equivale a $1.429 \text{ kWh}_e/\text{h}$.

Las variables adicionales del modelo son:

ctfrío: energía eléctrica equivalente utilizada para cargar el tanque en la hora t [kWh_e].

stfrío: energía eléctrica equivalente ahorrada al descargar el tanque en la hora t [kWh_e].

SOctfrío: estado de carga del tanque en la hora t , expresado en unidades eléctricas equivalentes [kWh_e].

ztfrío: variable binaria de operación del tanque. Toma valor 1 cuando el tanque puede cargarse y valor 0 cuando puede descargarse.

El balance eléctrico de la instalación queda definido como:

$$et_{red} = x_t + inft + comt + et_{frío} + ct_{frío} - st_{frío} \quad \forall t \in T$$

Ecuación 20 – Balance eléctrico del Modelo 6 con CTES

donde *etfrío* es el consumo eléctrico base de refrigeración modelado en el Modelo 4. La carga del tanque aumenta el consumo eléctrico del chiller en la hora correspondiente, mientras que la descarga lo reduce.

Para evitar que el modelo reduzca artificialmente el consumo de refrigeración por debajo de cero, se impone:

$$stfrío \leq etfrío + ctfrío \quad \forall t \in T$$

Ecuación 21 – Restricción de consumo no negativo del chiller

La evolución del estado de carga del tanque incorpora las pérdidas térmicas horarias y las eficiencias de carga y descarga:

$$SOCtfrío = (1 - \lambda) \cdot SOCt - 1frío + \eta_c \cdot ctfrío - stfrío / \eta_d \quad \forall t \in T$$

Ecuación 22 – Evolución del SOC del tanque térmico

donde $\lambda = 0,005$ corresponde a la pérdida térmica horaria del 0,5 %.

La operación del tanque está sujeta a los límites de capacidad y a las restricciones de no simultaneidad entre carga y descarga:

$$0 \leq SOCtfrío \leq 8.571 \quad \forall t \in T$$

Ecuación 23 – Límites del SOC del tanque térmico en unidades eléctricas equivalentes

$$ztfrío \in \{0,1\} \quad \forall t \in T$$

$$0 \leq ctfrío \leq 1.429 \cdot ztfrío \quad \forall t \in T$$

$$0 \leq stfrío \leq 1.429 \cdot (1 - ztfrío) \quad \forall t \in T$$

Ecuación 24 – Límites de potencia y no simultaneidad del CTES

Con estas restricciones, el tanque puede operar en modo carga o en modo descarga en cada hora, pero no en ambos simultáneamente. Por ello, el Modelo 6 se formula como un problema de programación lineal entera mixta (MILP).

Finalmente, se impone una condición cíclica anual para evitar que el modelo descargue artificialmente el tanque a lo largo del año:

$$SOC_{8760\text{frío}} \geq SOC_{0\text{frío}} = 2.571 \text{ kWh}$$

Ecuación 25 – Condición cíclica anual del tanque térmico

La función objetivo es equivalente a la de los modelos anteriores: minimizar el coste total de la energía eléctrica adquirida de la red:

$$\min \sum_{t \in T} e_{\text{red}} \cdot p_t$$

Ecuación 26 – Función objetivo del Modelo 6

5.8.4 RESULTADOS

El coste anual del Modelo 6 es de 7.918.526 €, con un ahorro energético de 985.350 € (11,07 %) respecto al caso base. El almacenamiento térmico permite desplazar parte del consumo eléctrico asociado a la refrigeración desde horas de precio elevado hacia horas de menor precio, aunque su efecto sobre el coste total es inferior al del BESS porque actúa únicamente sobre la demanda de refrigeración y no sobre el conjunto de la demanda eléctrica del CPD.

Además del ahorro energético, el CTES libera capacidad eléctrica asociada al sistema de refrigeración, que puede destinarse potencialmente a carga IT adicional. En el escenario económico considerado, este valor de capacidad IT liberada se estima en 198.476 €/año, calculado con un precio del compute de 120 €/MWh y un factor de uso del 60 % de la capacidad liberada, hipótesis cuya sensibilidad se analiza en la Tabla 12. Por tanto, el beneficio bruto anual del Modelo 6 asciende a 1.183.826 €/año (Figura 15).

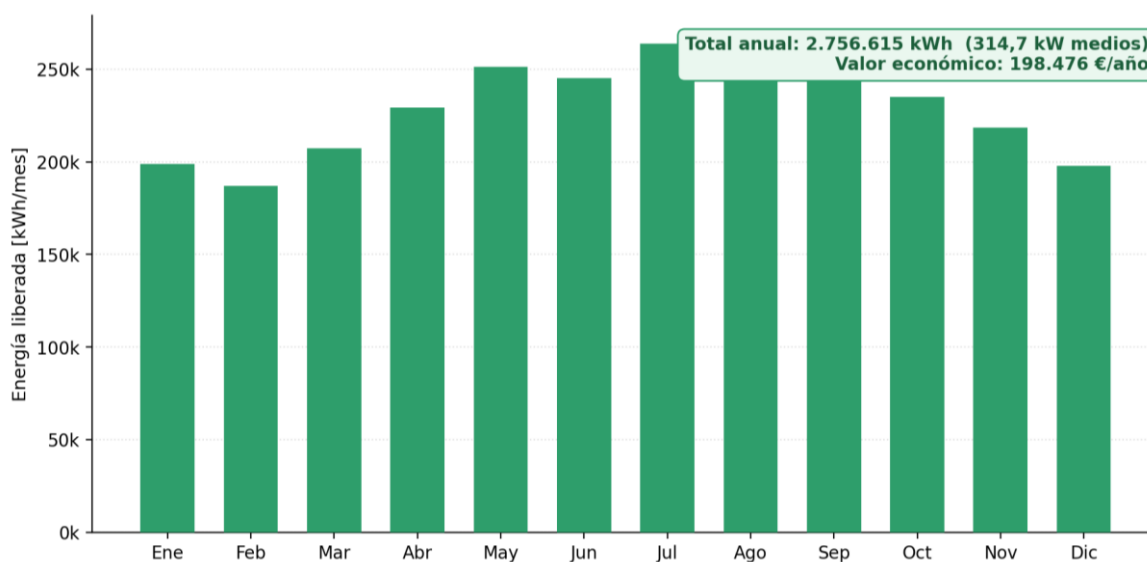


Figura 16. Capacidad eléctrica IT liberada por el CTES del Modelo 6

5.9 MODELO 7: SISTEMA HÍBRIDO CON BATERÍA BESS Y ALMACENAMIENTO TÉRMICO CTES

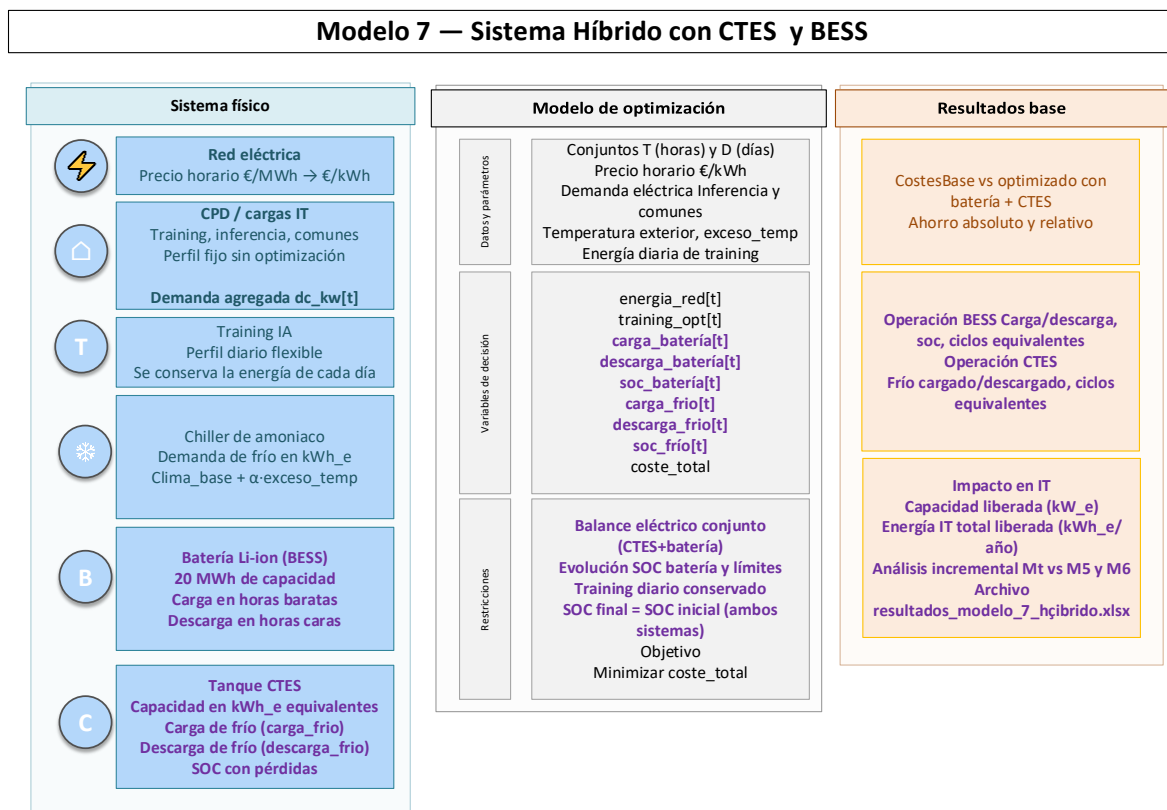


Figura 17 – Diagrama base Modelo 7

5.9.1 DESCRIPCIÓN Y MOTIVACIÓN

El Modelo 7 combina, sobre la base del Modelo 4, los dos sistemas de almacenamiento estudiados de forma independiente en los Modelos 5 y 6: la batería BESS de 20 MWh y el tanque de almacenamiento térmico CTES de 30 MWh_{th}. A diferencia de M5 y M6, que se plantean como alternativas tecnológicas entre sí sobre la misma base, el Modelo 7 evalúa si

la combinación de ambos activos aporta un valor superior al de cada uno por separado, y en qué medida el orden de incorporación de las tecnologías afecta a la rentabilidad marginal de cada inversión.

El optimizador coordina simultáneamente ambos almacenamientos junto con la programación flexible del entrenamiento: la batería arbitra sobre el conjunto de la demanda eléctrica del CPD, mientras que el tanque de frío desplaza específicamente el consumo eléctrico asociado a la refrigeración, liberando capacidad eléctrica para carga IT adicional, igual que en el Modelo 6.

5.9.2 PARÁMETROS DEL SISTEMA

Los parámetros de ambos subsistemas son idénticos a los empleados en los Modelos 5 y 6 de forma individual.

Tabla 5. Parámetros del sistema BESS en el Modelo 7 (idénticos al Modelo 5).

Parámetro	Valor
Capacidad nominal	20.000 kWh (20 MWh)
Potencia máxima de carga	5.000 kW
Potencia máxima de descarga	5.000 kW
Eficiencia de carga (η_c)	0,95
Eficiencia de descarga (η_d)	0,95
SOC inicial	10.000 kWh (50 % de capacidad)

Tabla 6. Parámetros del sistema CTES en el Modelo 7 (idénticos al Modelo 6).

Parámetro	Valor
Capacidad térmica nominal	30.000 kWh _{th} (≈ 8.571 kWh _e con COP = 3,5)
Potencia máxima de carga	5.000 kWh _{th} /h (≈ 1.429 kWh _e /h)
Potencia máxima de descarga	5.000 kWh _{th} /h (≈ 1.429 kWh _e /h)
Pérdida térmica horaria	0,5 %
Eficiencia de carga (η_c)	0,95
Eficiencia de descarga (η_d)	0,95
SOC inicial	2.571 kWh _e (30 % de la capacidad)
COP del chiller	3,5

El coste de inversión total del Modelo 7 asciende a 13.400.000 €, resultado de sumar el chiller de amoníaco (4.500.000 €, compartido y contabilizado una sola vez), la batería ($20.000 \text{ kWh} \times 300 \text{ €/kWh} = 6.000.000 \text{ €}$), el tanque CTES ($30.000 \text{ kWh}_{th} \times 80 \text{ €/kWh}_{th} = 2.400.000 \text{ €}$) y los costes de integración (500.000 €, también contabilizados una única vez). El CAPEX del Modelo 7 (13.400.000 €) es inferior a la suma directa de los CAPEX de M5 y M6 por separado ($11.000.000 \text{ €} + 7.400.000 \text{ €} = 18.400.000 \text{ €}$): la diferencia de 5.000.000 € corresponde al chiller de amoníaco y a los costes de integración, que solo se pagan una vez al compartir infraestructura entre ambos sistemas de almacenamiento. El OPEX anual asciende a 234.000 €/año (2 % del CAPEX del chiller + 2 % del CAPEX de la batería + 1 % del CAPEX del tanque).

5.9.3 VARIABLES ADICIONALES

Las variables del Modelo 7 combinan las de los Modelos 5 y 6:

- $c_bat(t)$, $s_bat(t)$, $SOC_bat(t)$, $z_bat(t)$: carga, descarga, estado de carga y variable binaria de operación de la batería (idénticas al Modelo 5).
- $c_frio(t)$, $s_frio(t)$, $SOC_frio(t)$, $z_frio(t)$: carga, descarga, estado de carga y variable binaria de operación del tanque térmico, en unidades eléctricas equivalentes (idénticas al Modelo 6).

5.9.4 FORMULACIÓN MATEMÁTICA COMPLETA

Balance eléctrico: en cada hora, la energía comprada a la red debe cubrir la demanda total del CPD (training, inferencia, comunes y refrigeración) más el neto de ambos sistemas de almacenamiento:

$$e_red(t) = training_opt(t) + inferencia(t) + comunes(t) + demanda_frio(t) + c_bat(t) - s_bat(t) + c_frio(t) - s_frio(t), \forall t \in T$$

Ecuación 27 – Balance eléctrico del Modelo 7 con BESS y CTES

Las restricciones de evolución del SOC, límites de potencia y no simultaneidad de cada sistema son idénticas a las formuladas en los Modelos 5 (batería) y 6 (tanque térmico), aplicadas de forma simultánea. Se mantienen también las condiciones cíclicas anuales de ambos sistemas (SOC final = SOC inicial), y la restricción de no negatividad del consumo del chiller.

Por incluir dos pares de variables binarias (una por sistema de almacenamiento, en cada hora), el Modelo 7 se formula como un problema de programación lineal entera mixta (MILP) de mayor tamaño que los Modelos 5 y 6 por separado.

La función objetivo es equivalente a la de los modelos anteriores: minimizar el coste total de la energía eléctrica adquirida de la red:

$$\min \sum_{t \in T} e_{red}(t) \cdot p(t)$$

Ecuación 28 – Función objetivo del Modelo 7

5.9.5 RESULTADOS

El coste anual del Modelo 7 es de 7.302.797 €, con un ahorro energético de 1.601.078 € (17,98 %) respecto al caso base, el mayor ahorro porcentual de todos los modelos evaluados, superior incluso al del Modelo 2. Sumando el valor de la capacidad IT liberada por el CTES (198.411 €/año, prácticamente idéntico al del Modelo 6 y calculado con las mismas hipótesis: precio del compute de 120 €/MWh y factor de uso del 60 %), el beneficio bruto anual asciende a 1.799.489 €/año.

Sin embargo, la combinación de ambas tecnologías no se traduce en una mejora proporcional de la rentabilidad: el VAN del Modelo 7 (422.196 €) es sensiblemente inferior al del Modelo 6 en solitario (1.990.431 €), con una TIR del 8,49 % frente al 12,05 % de M6 y un Índice de Rentabilidad de 1,032, frente a 1,269.

Para entender esta diferencia se ha realizado un análisis incremental, evaluando el efecto marginal de añadir cada tecnología sobre la otra ya instalada:

- Añadir la batería sobre un CTES ya instalado (M6 → M7): requiere un CAPEX adicional de 6.000.000 € y aporta un beneficio incremental de 615.663 €/año, lo que arroja un VAN incremental de -1.568.235 € y una TIR del 3,6 %. La inversión marginal en batería no se justifica una vez instalado el CTES.
- Añadir el CTES sobre una batería ya instalada (M5 → M7): requiere un CAPEX adicional de 2.400.000 € y aporta un beneficio incremental de 359.910 €/año, con un VAN incremental de 550.874 € y una TIR del 11,5 %. Esta inversión marginal sí es rentable, y de hecho es la que permite que el sistema combinado alcance un VAN

positivo, corrigiendo la inviabilidad marginal del Modelo 5 en solitario ($VAN = -128.678 \text{ €}$).

Estos resultados indican que el orden de incorporación de las tecnologías no es indiferente: añadir el CTES a una instalación que ya dispone de batería es una ampliación rentable, mientras que, una vez instalado el CTES, la incorporación posterior de la batería no se justifica económicamente con las hipótesis adoptadas. El Modelo 6 se mantiene como la recomendación principal por rentabilidad, mientras que el Modelo 7 se presenta como la alternativa de máximo ahorro para escenarios donde la disponibilidad de capital no es la restricción principal.

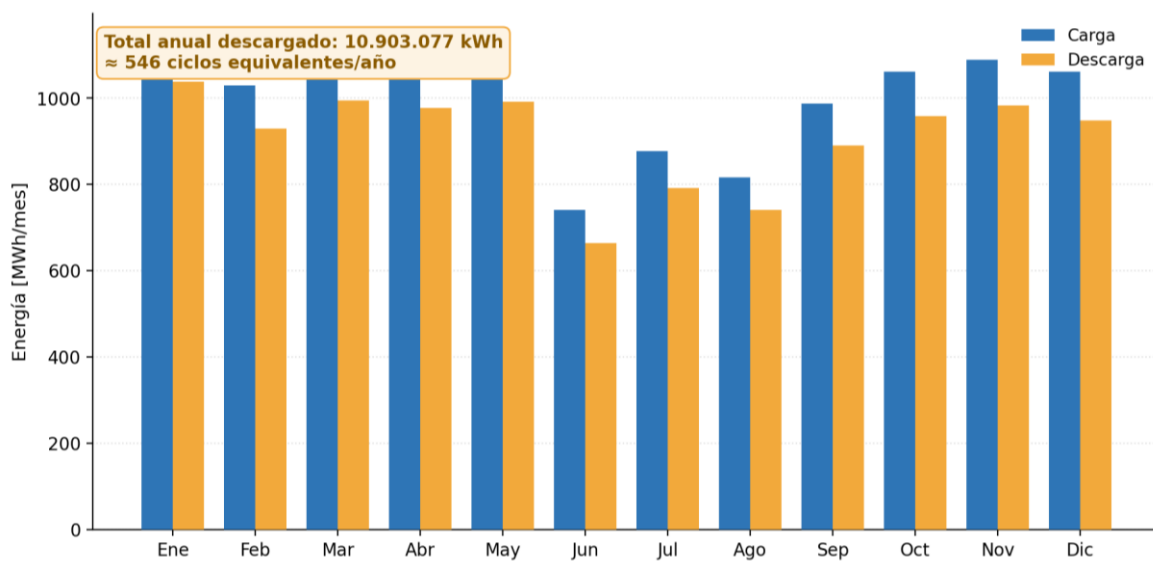


Figura 18. Energía gestionada por la batería BESS en el Modelo 7

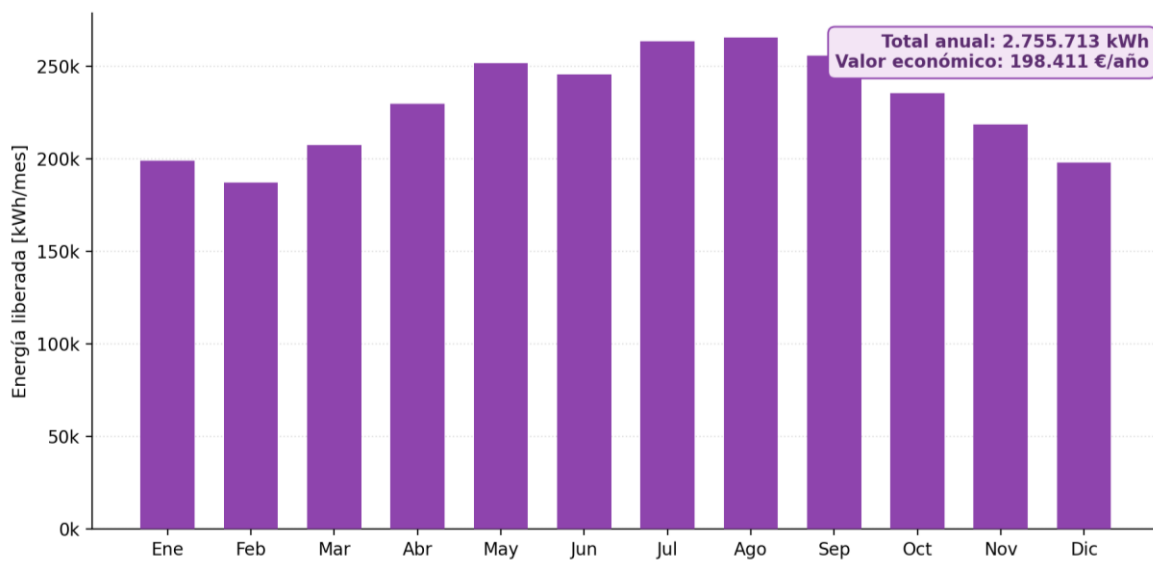


Figura 19. Capacidad eléctrica de IT liberada en el Modelo 7

Capítulo 6. ANÁLISIS DE RESULTADOS

Análisis económico y de sensibilidad — Comparativa M1 a M7		
Sistemas modelados	Métricas financieras	Sensibilidad y viabilidad
M1 - Modelo base de cargas IT y red eléctrica Con precio horario, sin flexibilidad de training ni almacenamiento; sirve de referencia del comportamiento del CPD con cargas fijas de inferencia, climatización y consumos comunes.	VAN Valor actual neto del proyecto	4 tablas de sensibilidad Enfocadas en M6
M2 - Flexibilidad de training anual Mantiene constante la energía anual de training IA, pero permite reprogramarla a lo largo del año para aprovechar horas baratas, respetando el límite de potencia por hora.	TIR Rentabilidad interna anual	Tornado chart Impacto de parámetros en VAN M6
M3 - Flexibilidad de training diaria Conserva la energía de training día a día y la redistribuye dentro del mismo día hacia las horas más baratas, con tope horario, igual que en los modelos anteriores.	Payback simple y descontado Tiempo de recuperación	Breakeven analysis Umbral de viabilidad M5 y M6
M4 - Modelo climático y refrigeración por amoniac Introduce un modelo de temperatura exterior y su impacto en la refrigeración del CPD, acoplando la climatización al perfil horario de temperatura y precio, con frío producido por chiller de amoniac.	PI Índice de rentabilidad	Análisis complementario M6 solo ahorro energético · M6 incremental vs M4 M7 incremental vs M5 y vs M6
M5 - Sistema con BESS Añade baterías Li-ion (BESS) que se cargan en horas baratas y se descargan en horas caras, integrando cargas IT, red y batería para minimizar el coste y mejorar la gestión de capacidad.	LCOE desglosado Capital + O&M + Energía IEA / OCDE	Conclusión esperada M6: mejor retorno por euro invertido (ahorro + capacidad IT + robustez financiera) M7: máximo ahorro absoluto combinando tecnologías, con rentabilidad marginal decreciente
M6 - CTES acoplado al chiller de amoniac y a las cargas de training IA Modelo secuencial horario que gestiona el estado de carga del tanque, el uso del chiller y la programación del training para minimizar el coste y liberar capacidad de IT.	Horizonte: 15 años WACC: 8%	Salida analisis_economico_tfg.xlsx
M7 - Sistema híbrido BESS + CTES Combina batería y tanque térmico sobre la base del Modelo 4, coordinando ambos almacenamientos con el training flexible para maximizar el ahorro y evaluar la rentabilidad marginal de combinarlos.		

Figura 20 – Diagrama base Análisis de resultados

6.1 COMPARACIÓN DE COSTES ENTRE MODELOS

6.1.1 COSTES ANUALES Y AHORROS

La Tabla 7 recoge los resultados económicos de los siete modelos comparados con el caso base, ordenados por nivel de ahorro alcanzado.

Tabla 7. Comparativa de costes anuales, ahorros y CAPEX por modelo.

Modelo	Coste anual (€)	Ahorro anual (€)	Ahorro (%)	CAPEX (€)
M1: Caso base	8.903.875	n/a	n/a	0
M4: Clima + flex.	8.081.904	821.972	9,23 %	4.500.000
M3: Flex. diaria	8.080.644	823.231	9,25 %	0
M6: CTES + clima	7.918.526	985.350	11,07 %	7.400.000
M5: Batería + clima	7.464.296	1.439.579	16,17 %	11.000.000
M2: Flex. anual	7.358.840	1.545.035	17,35 %	0
M7: Híbrido BESS+CTES	7.302.797	1.601.078	17,98 %	13.400.000

6.1.2 ANÁLISIS DE LOS RESULTADOS POR MODELO

Modelo 2 frente a Modelo 3. La diferencia más llamativa del análisis es el salto entre el ahorro del Modelo 2 (flexibilidad anual, 17,35 %) y el del Modelo 3 (flexibilidad diaria, 9,25 %). Esos 721.804 €/año de diferencia ponen cifra al valor de la flexibilidad inter-diaria, es decir, a la posibilidad de concentrar varias semanas de training en los pocos días más baratos del año. El Modelo 2 puede aprovechar esa libertad, mientras que el Modelo 3 no. En la práctica esa libertad no es operativamente viable (los trabajos de entrenamiento tienen plazos de ejecución bastante acotados), así que el Modelo 3 marca el límite realista del ahorro que se puede conseguir gestionando solo la carga de training, sin inversión de capital.

Modelo 4 frente a Modelo 3. Los Modelos 3 y 4 arrojan costes prácticamente iguales: apenas 1.260 €/año los separan. Lo que esto indica es que sustituir el perfil de climatización del dataset por el modelo térmico calibrado no altera la estructura del problema de optimización, y eso valida la coherencia entre ambas representaciones. El sistema de refrigeración modelado con amoníaco reproduce bien el comportamiento del consumo real

de climatización, pero por sí solo no aporta ahorros frente al Modelo 3, porque la demanda de refrigeración se sigue tratando en el Modelo 4 como parámetro fijo y no como una variable de decisión más.

Modelo 5 frente a Modelo 4. Al incorporar la batería se añaden 617.608 €/año de ahorro sobre el Modelo 4, lo que eleva el ahorro total hasta 1.439.579 €/año (16,17 %) respecto al caso base. El mecanismo de valor es el arbitraje horario: la batería carga en horas de menor precio y descarga en horas de mayor precio, aprovechando la volatilidad de OMIE en 2025. La formulación incorpora una variable binaria que impide la carga y descarga simultáneas, por lo que el Modelo 5 se resuelve como un problema de programación lineal entera mixta (MILP).

Modelo 6 frente a Modelo 5. El Modelo 6 alcanza un ahorro energético de 985.350 €/año (11,07 %), inferior al del Modelo 5, porque el arbitraje térmico actúa solo sobre la demanda de refrigeración y no sobre toda la energía eléctrica del CPD. A cambio, su CAPEX es un 32,7 % inferior (7,4 M€ frente a 11,0 M€), no requiere reemplazo dentro del horizonte de 15 años y libera capacidad eléctrica asociada a la reducción del consumo del chiller, valorada en 198.476 €/año. Al sumar este valor al ahorro energético, el beneficio bruto anual del Modelo 6 asciende a 1.183.826 €/año. El detalle de su rentabilidad se recoge en la Tabla 9.

Modelo 7 frente a los Modelos 5 y 6. El Modelo 7 combina ambos almacenamientos y alcanza el menor coste anual de todos los modelos (7.302.797 €), con un ahorro energético de 1.601.078 €/año (17,98 %), superior incluso a la cota teórica del Modelo 2. Sin embargo, la sinergia es solo parcial: el ahorro conjunto queda muy por debajo de la suma de los ahorros individuales de M5 y M6 (2.424.929 €/año), porque ambos sistemas compiten por las mismas horas baratas del mercado y el arbitraje de la batería ya cubre parte de la demanda de refrigeración que desplazaría el tanque. Sumando la capacidad IT liberada por el CTES (198.411 €/año, prácticamente idéntica a la del Modelo 6), el beneficio bruto anual asciende a 1.799.489 €/año, el mayor de todo el estudio.

6.2 ANÁLISIS ECONÓMICO DE LA INVERSIÓN

6.2.1 HIPÓTESIS DE CAPEX Y OPEX

Los Modelos 2 y 3 no exigen inversión de capital, porque solo introducen cambios en la estrategia de despacho de la carga de training, así que su rentabilidad es inmediata e incondicional. Los Modelos 4 a 7, en cambio, sí asumen la instalación de infraestructura nueva, cuyo coste se recoge en la Tabla 8.

Tabla 8. Desglose de CAPEX por componente.

Componente	Hipótesis	CAPEX (€)
Batería BESS	20.000 kWh $\times$ 300 €/kWh	6.000.000
Sistema de refrigeración amoniaco	5.000 kW $\times$ 900 €/kW	4.500.000
Costes de integración y control	Estimación fija	500.000
CAPEX total (M5)		11.000.000
CAPEX M4 (solo refrigeración)		4.500.000
Tanque inercial CTES	30.000 kWh _{th} $\times$ 80 €/kWh _{th}	2.400.000
CAPEX total (M6)		7.400.000
CAPEX total (M7, todos los activos)		13.400.000

Los costes anuales de operación y mantenimiento se han estimado como un porcentaje del CAPEX de cada activo. En el Modelo 5, un 2,0 % de la batería (120.000 €/año) y un 2,0 % del sistema de amoniaco (90.000 €/año) dan un OPEX de 210.000 €/año, que reduce el beneficio bruto hasta un ahorro neto de 1.229.579 €/año. En el Modelo 6, un 2,0 % del chiller de amoniaco (90.000 €/año) y un 1,0 % del tanque CTES (24.000 €/año) suman un OPEX

de 114.000 €/año, con un ahorro neto de 1.069.826 €/año. En el Modelo 7, que agrupa todos los activos (2,0 % de la batería, 2,0 % del chiller de amoníaco y 1,0 % del tanque CTES), el OPEX asciende a 234.000 €/año, con un ahorro neto de 1.565.489 €/año sobre el beneficio bruto.

Las hipótesis financieras adoptadas son: tasa de descuento del 8 %, horizonte de evaluación de 15 años, crecimiento del ahorro del 0 % anual (escenario conservador) y valor residual del 10 % del CAPEX al final del horizonte.

6.2.2 VAN, TIR Y PERIODO DE RETORNO

La Tabla 9 compara los indicadores de rentabilidad de los Modelos 5, 6 y 7.

Tabla 9. Indicadores de rentabilidad de los Modelos 5, 6 y 7.

Indicador	Modelo 5	Modelo 6	Modelo 7
Beneficio bruto anual	1.439.579 €	1.183.826 €	1.799.489 €
OPEX anual	210.000 €	114.000 €	234.000 €
Ahorro neto anual	1.229.579 €	1.069.826 €	1.565.489 €
CAPEX total	11.000.000 €	7.400.000 €	13.400.000 €
Payback simple	9,0 años	7,0 años	9,0 años
Payback descontado	No recupera	11,0 años	15,0 años
VAN (8 %, 15 años)	-128.678 €	+1.990.431 €	+422.196 €
TIR	7,82 %	12,05 %	8,49 %
Índice de rentabilidad (PI)	0,99	1,27	1,03

La conclusión del análisis económico es que, con las hipótesis de base adoptadas, la inversión en el sistema con BESS queda al borde de la viabilidad: el VAN es ligeramente negativo (-128.678 €) y la TIR (7,82 %) se sitúa prácticamente en el entorno de la tasa de descuento exigida (8 %). El payback simple de 9 años queda dentro del horizonte de evaluación de 15 años, aunque el payback descontado no se alcanza en el escenario base.

El Modelo 6, en cambio, resulta viable con holgura. Su VAN asciende a 1.990.431 €, la TIR alcanza el 12,05 % y el periodo de retorno descontado se sitúa en 11 años, con un índice de rentabilidad de 1,27. La ventaja del CTES no se basa en lograr un mayor ahorro energético que el BESS, sino en combinar un CAPEX menor, ausencia de reemplazo y valor adicional de capacidad IT liberada.

El Modelo 7 combina ambos almacenamientos y logra el mayor ahorro neto anual (1.565.489 €/año), pero su CAPEX conjunto de 13.400.000 € modera la rentabilidad: el VAN se reduce a +422.196 €, la TIR al 8,49 % y el payback descontado se sitúa en 15 años, justo en el límite del horizonte de evaluación, con un índice de rentabilidad de 1,03. La combinación es, por tanto, viable pero claramente menos eficiente por euro invertido que el CTES en solitario; el análisis incremental de esta comparativa se desarrolla en la sección siguiente.

Entre las alternativas sin inversión, el Modelo 3 sigue siendo el más eficiente, al generar 823.231 €/año de ahorro sin necesidad de invertir capital. Cuando sí se contempla invertir capital, el Modelo 6 es la opción con mejor rentabilidad en el escenario base, el Modelo 5 queda en el umbral de viabilidad y el Modelo 7, viable pero con menor retorno por euro invertido, se reserva para escenarios donde prime maximizar el ahorro absoluto.

La figura siguiente muestra cómo se reparte el coste a lo largo del día para cada modelo, sumando todas las horas del año. Se puede ver que todos los modelos concentran el gasto en las mismas franjas horarias, coincidiendo con las horas de mayor demanda del sistema eléctrico, aunque con distinta intensidad según cuánta flexibilidad o almacenamiento tiene cada uno para esquivar esos picos.

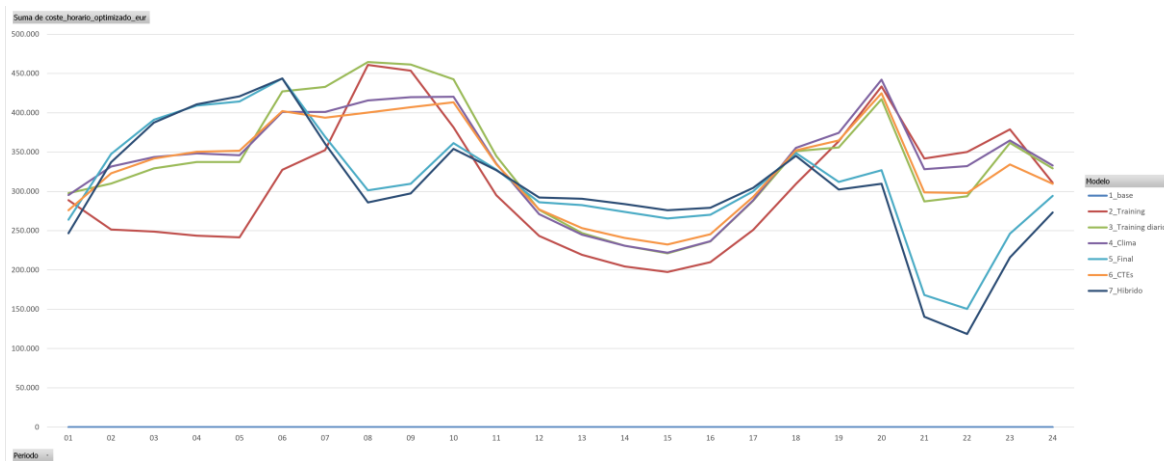


Figura 21. Coste horario optimizado agregado por modelo

6.2.3 COMPARATIVA DE TECNOLOGÍAS DE ALMACENAMIENTO: BESS, CTES E HÍBRIDO

De los siete modelos desarrollados, M5, M6 y M7 son los únicos que representan alternativas de inversión mutuamente comparables sobre la misma base (Modelo 4): todos añaden un sistema de almacenamiento, con distinto coste y mecanismo de generación de valor. La Figura 21 resume de un vistazo la decisión: el eje horizontal muestra la inversión requerida (CAPEX); el eje vertical, la rentabilidad generada (VAN a 15 años); y el tamaño de la burbuja, el beneficio bruto anual.

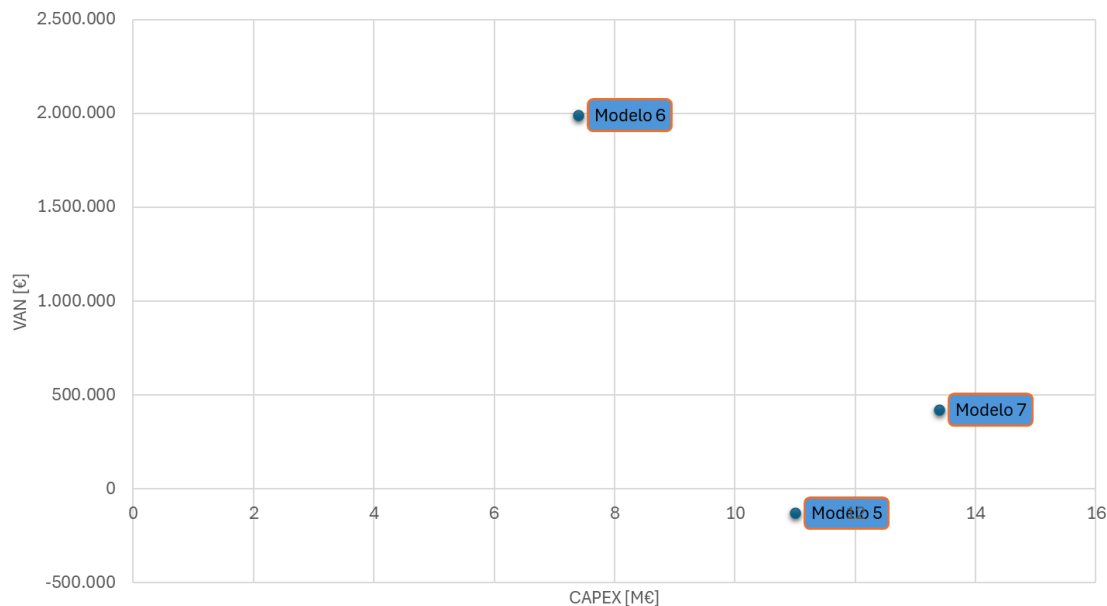


Figura 22. Decisión de inversión entre BESS, CTES e híbrido: CAPEX frente a VAN

El Modelo 6 (CTES) domina claramente en términos de retorno por euro invertido: con el CAPEX más bajo de los tres (7.400.000 €), alcanza el VAN más alto (1.990.431 €), la TIR más alta (12,05 %) y el periodo de recuperación más corto (7 años simple, 11 descontado). El Modelo 5 (BESS en solitario) es la peor alternativa de las tres: requiere el mayor desembolso relativo a su beneficio y su VAN es negativo (-128.678 €), por lo que no se recomienda como inversión aislada.

El Modelo 7 (híbrido) ofrece el mayor ahorro absoluto de todos los modelos desarrollados (1.799.489 €/año) pero, al requerir el CAPEX más elevado (13.400.000 €), su rentabilidad relativa es intermedia (VAN 422.196 €, TIR 8,49 %), muy por debajo de la que ofrece el CTES en solitario. El análisis incremental confirma que esta diferencia depende del orden de incorporación de las tecnologías: añadir la batería sobre un CTES ya instalado no es rentable (VAN incremental -1.568.235 €), mientras que añadir el CTES sobre una batería ya instalada sí lo es (VAN incremental +550.874 €); esta segunda secuencia es la que rescata al Modelo 5 de su inviabilidad en solitario.

La recomendación de este trabajo es, por tanto, doble y depende del criterio de decisión del promotor: si el objetivo es maximizar el retorno por euro invertido, el Modelo 6 (CTES) es la alternativa recomendada. Si el objetivo es maximizar el ahorro energético absoluto y la disponibilidad de capital no es la restricción principal, el Modelo 7 (híbrido) es la alternativa preferible. En ningún escenario se recomienda la instalación de la batería BESS como sistema de almacenamiento único.

6.3 ANÁLISIS DE SENSIBILIDAD

El análisis de sensibilidad cubre los tres modelos que incorporan almacenamiento (M5, M6 y M7) con un enfoque distinto para cada uno. El Modelo 5 queda al borde de la viabilidad en el escenario base (VAN -128.678 €), por lo que el objetivo es identificar qué variación mínima de precio, CAPEX o tasa de descuento lo convertiría en una inversión rentable. Los Modelos 6 y 7, en cambio, sí son viables en el escenario base y representan los dos criterios de decisión planteados en el apartado anterior: máxima rentabilidad por euro invertido (M6) y máximo ahorro absoluto (M7); aquí el objetivo es comprobar hasta qué punto se mantiene esa viabilidad frente a los parámetros más inciertos.

6.3.1 SENSIBILIDAD DEL MODELO 5 AL WACC, AL CRECIMIENTO DEL PRECIO DE LA ENERGÍA Y AL CAPEX

Tabla M5-A. Sensibilidad del VAN de M5 (€) al WACC y al crecimiento anual del precio de la energía.

WACC	Crec. 0 %	Crec. 1 %	Crec. 2 %	Crec. 3 %	Crec. 4 %
4 %	3.281.727	4.327.990	5.464.081	6.698.153	8.039.088
5 %	2.291.728	3.240.690	4.269.930	5.386.670	6.598.776
6 %	1.400.969	2.263.443	3.197.787	4.210.411	5.308.290
7 %	597.590	1.383.039	2.232.943	3.153.000	4.149.405
8 %	-128.679	588.039	1.362.664	2.200.265	3.106.356
10 %	-1.384.393	-784.151	-136.936	561.289	1.314.906
12 %	-2.424.538	-1.918.065	-1.373.236	-786.819	-155.308

Tabla M5-B. Sensibilidad del VAN de M5 (€) al beneficio total y al CAPEX.

Variación beneficio	CAPEX -20 %	CAPEX -10 %	CAPEX +0 %	CAPEX +10 %	CAPEX +20 %
Beneficio -20 %	-462.441	-1.527.765	-2.593.088	-3.658.411	-4.723.735
Beneficio -10 %	769.763	-295.560	-1.360.883	-2.426.207	-3.491.530
Beneficio +0 %	2.001.968	936.645	-128.679	-1.194.002	-2.259.326
Beneficio +10 %	3.234.173	2.168.849	1.103.526	38.202	-1.027.121
Beneficio +20 %	4.466.377	3.401.054	2.335.730	1.270.407	205.083

El Modelo 5 resulta especialmente sensible: con un crecimiento del precio de la energía de tan solo un 0,2 % anual, el VAN pasa a ser positivo incluso sin cambiar ninguna otra hipótesis. De forma equivalente, basta una caída del CAPEX de la batería de apenas un 1,3 % (de 300 €/kWh a unos 293 €/kWh) para que la inversión sea rentable en el escenario base. Esta cercanía al umbral confirma que el BESS en solitario se sitúa justo en el límite de la rentabilidad, a diferencia de M6 y M7, que necesitan variaciones mucho mayores en sus parámetros para dejar de ser rentables.

6.3.2 SENSIBILIDAD AL COSTE DEL TANQUE Y AL BENEFICIO TOTAL ANUAL

La Tabla 10 muestra cómo varía el VAN del Modelo 6 ante cambios simultáneos en el coste unitario del tanque CTES y en el beneficio total anual. El caso base corresponde a 80 €/kWh_{th} y beneficio +0 %, con un VAN de 1.990.431 €.

Tabla 10. Sensibilidad del VAN de M6 (€) al coste unitario del tanque y al beneficio total anual.

Coste tanque	Beneficio -20 %	Beneficio -10 %	Beneficio +0 %	Beneficio +10 %	Beneficio +20 %
40 €/kWh _{th}	1.126.016	2.139.309	3.152.602	4.165.895	5.179.189
60 €/kWh _{th}	544.930	1.558.223	2.571.517	3.584.810	4.598.103
80 €/kWh _{th}	-36.155	977.138	1.990.431	3.003.724	4.017.018
100 €/kWh _{th}	-617.241	396.052	1.409.346	2.422.639	3.435.932
120 €/kWh _{th}	-1.198.326	-185.033	828.260	1.841.553	2.854.847
140 €/kWh _{th}	-1.779.412	-766.119	247.175	1.260.468	2.273.761

6.3.3 SENSIBILIDAD AL WACC Y AL CRECIMIENTO DEL PRECIO DE LA ENERGÍA

La Tabla 11 analiza la sensibilidad del VAN frente a la tasa de descuento y al crecimiento anual del precio de la energía. Con WACC del 8 % y crecimiento nulo, el VAN es de

1.990.431 €. La rentabilidad mejora de forma significativa si se considera un crecimiento del precio eléctrico, mientras que tasas de descuento más altas reducen el margen económico.

Tabla 11. Sensibilidad del VAN de M6 (€) al WACC y al crecimiento anual del precio de la energía.

WACC	Crec. 0 %	Crec. 1 %	Crec. 2 %	Crec. 3 %	Crec. 4 %
4 %	4.905.635	5.766.021	6.700.276	7.715.105	8.817.812
5 %	4.060.380	4.840.751	5.687.137	6.605.480	7.602.245
6 %	3.299.192	4.008.440	4.776.790	5.609.514	6.512.345
7 %	2.612.093	3.258.000	3.956.911	4.713.512	5.532.898
8 %	1.990.431	2.579.818	3.216.824	3.905.619	4.650.735
10 %	914.331	1.407.935	1.940.168	2.514.347	3.134.077
12 %	21.635	438.129	886.164	1.368.399	1.887.717

6.3.4 SENSIBILIDAD AL VALOR DE LA CAPACIDAD IT LIBERADA

La Tabla 12 recoge la sensibilidad del VAN frente al precio asignado al compute adicional y al factor de uso de la capacidad IT liberada. Esta hipótesis es relevante porque el CTES, además de reducir el coste energético de refrigeración, libera potencia eléctrica que podría destinarse a carga IT adicional.

Tabla 12. Sensibilidad del VAN de M6 (€) al precio del compute y al factor de uso de la capacidad IT liberada.

Precio compute	Factor 30 %	Factor 40 %	Factor 50 %	Factor 60 %	Factor 70 %	Factor 80 %
60 €/MWh	716.291	857.862	999.433	1.141.004	1.282.575	1.424.147
80 €/MWh	857.862	1.046.624	1.235.385	1.424.147	1.612.908	1.801.670

Precio compute	Factor 30 %	Factor 40 %	Factor 50 %	Factor 60 %	Factor 70 %	Factor 80 %
100 €/MWh	999.433	1.235.385	1.471.337	1.707.289	1.943.241	2.179.193
120 €/MWh	1.141.004	1.424.147	1.707.289	1.990.431	2.273.573	2.556.716
150 €/MWh	1.353.361	1.707.289	2.061.217	2.415.145	2.769.072	3.123.000
200 €/MWh	1.707.289	2.179.193	2.651.096	3.123.000	3.594.904	4.066.808

6.3.5 SENSIBILIDAD AL BENEFICIO TOTAL Y AL CAPEX DEL M6

La Tabla 13 muestra el efecto combinado de variar el beneficio total anual y el CAPEX del Modelo 6. El caso base se sitúa en el centro de la tabla, con un VAN de 1.990.431 €.

Tabla 13. Sensibilidad del VAN de M6 (€) al beneficio total y al CAPEX.

Variación beneficio	CAPEX -20 %	CAPEX -10 %	CAPEX +0 %	CAPEX +10 %	CAPEX +20 %
Beneficio -20 %	1.397.189	680.517	-36.155	-752.828	-1.469.500
Beneficio -10 %	2.410.482	1.693.810	977.138	260.466	-456.206
Beneficio +0 %	3.423.775	2.707.103	1.990.431	1.273.759	557.087
Beneficio +10 %	4.437.069	3.720.396	3.003.724	2.287.052	1.570.380
Beneficio +20 %	5.450.362	4.733.690	4.017.018	3.300.346	2.583.673

Las Tablas 10b a 13b repiten el mismo análisis para el Modelo 7, con idéntica metodología y los mismos rangos de variación, para poder comparar directamente el margen de seguridad de cada alternativa.

Tabla 10b. Sensibilidad del VAN de M7 (€) al coste unitario del tanque y al beneficio total anual.

Coste tanque	Beneficio -20 %	Beneficio -10 %	Beneficio +0 %	Beneficio +10 %	Beneficio +20 %
40 €/kWh_th	-1.496.173	44.096	1.584.365	3.124.633	4.664.902
60 €/kWh_th	-2.077.258	-536.990	1.003.279	2.543.548	4.083.817
80 €/kWh_th	-2.658.344	-1.118.075	422.194	1.962.462	3.502.731
100 €/kWh_th	-3.239.429	-1.699.161	-158.892	1.381.377	2.921.646
120 €/kWh_th	-3.820.515	-2.280.246	-739.977	800.291	2.340.560

Coste tanque	Beneficio -20 %	Beneficio -10 %	Beneficio +0 %	Beneficio +10 %	Beneficio +20 %
140 €/kWh_th	-4.401.600	-2.861.332	-1.321.063	219.206	1.759.475

Tabla 11b. Sensibilidad del VAN de M7 (€) al WACC y al crecimiento anual del precio de la energía.

WACC	Crec. 0 %	Crec. 1 %	Crec. 2 %	Crec. 3 %	Crec. 4 %
4 %	4.749.768	6.057.608	7.477.733	9.020.337	10.696.520
5 %	3.493.803	4.680.016	5.966.576	7.362.513	8.877.659
6 %	2.363.554	3.441.656	4.609.596	5.875.387	7.247.747
7 %	1.344.017	2.325.836	3.388.225	4.538.306	5.783.823
8 %	422.194	1.318.099	2.286.388	3.333.398	4.466.021
10 %	-1.171.981	-421.672	387.354	1.260.142	2.202.171
12 %	-2.492.854	-1.859.757	-1.178.715	-445.687	343.708

Tabla 12b. Sensibilidad del VAN de M7 (€) al precio del compute y al factor de uso de la capacidad IT liberada.

Precio compute	Factor 30 %	Factor 40 %	Factor 50 %	Factor 60 %	Factor 70 %	Factor 80 %
60 €/MWh	-851.527	-710.003	-568.478	-426.954	-285.429	-143.905
80 €/MWh	-710.003	-521.303	-332.604	-143.905	44.795	233.494
100 €/MWh	-568.478	-332.604	-96.730	139.144	375.019	610.893
120 €/MWh	-426.954	-143.905	139.144	422.194	705.243	988.292
150 €/MWh	-214.667	139.144	492.956	846.767	1.200.579	1.554.390
200 €/MWh	139.144	610.893	1.082.642	1.554.390	2.026.139	2.497.887

Tabla 13b. Sensibilidad del VAN de M7 (€) al beneficio total y al CAPEX.

Variación beneficio	CAPEX -20 %	CAPEX -10 %	CAPEX +0 %	CAPEX +10 %	CAPEX +20 %
Beneficio -20 %	-62.829	-1.360.586	-2.658.344	-3.956.102	-5.253.859
Beneficio -10 %	1.477.440	179.682	-1.118.075	-2.415.833	-3.713.590
Beneficio +0 %	3.017.709	1.719.951	422.194	-875.564	-2.173.322
Beneficio +10 %	4.557.978	3.260.220	1.962.462	664.705	-633.053
Beneficio +20 %	6.098.246	4.800.489	3.502.731	2.204.974	907.216

El análisis de sensibilidad muestra que el Modelo 6 mantiene un VAN positivo en el escenario base, aunque su rentabilidad depende especialmente del beneficio anual capturado, del coste unitario del tanque y de la valoración económica de la capacidad IT liberada. Incluso bajo una hipótesis conservadora sin ingresos adicionales por compute, el análisis complementario arroja un VAN positivo de 291.577 €, una TIR del 8,62 % y un índice de rentabilidad de 1,04, por lo que el CTES se mantiene ligeramente por encima del umbral de viabilidad. En términos incrementales frente al Modelo 4, considerando solo el tanque y la integración, el VAN es de 83.277 €, con una TIR del 8,45 %.

El análisis de sensibilidad del Modelo 7 muestra un margen de seguridad considerablemente más estrecho que el del Modelo 6. Mientras que el CAPEX de M6 puede encarecerse hasta un 28% antes de que el VAN se vuelva negativo, el de M7 solo admite un 3% de sobrecoste antes de cruzar ese umbral (13,83 M€ frente a los 13,4 M€ del escenario base). De forma análoga, M6 mantiene un VAN positivo incluso con una tasa de descuento del 10%, mientras que M7 ya resulta inviable a partir de esa misma tasa si no hay crecimiento del precio de la energía. Esto no invalida al híbrido como alternativa, ya que su escenario base sigue siendo viable y ofrece el mayor ahorro absoluto del estudio, pero sí indica que su viabilidad depende de un conjunto más amplio de hipótesis sostenidas simultáneamente, al tratarse de la inversión de mayor envergadura de las siete evaluadas.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1 CONCLUSIONES

Este trabajo ha desarrollado y comparado siete modelos de optimización del suministro energético para un centro de datos dedicado a inteligencia artificial de 20 MW de carga IT, conectado al mercado eléctrico mayorista español (OMIE) y operado durante el año 2025. El análisis parte de un caso base sin gestión activa y añade, modelo a modelo, nuevas palancas de flexibilidad, midiendo en cada etapa su impacto económico sobre el coste anual de energía.

7.1.1 PRIMERA CONCLUSIÓN: LA FLEXIBILIDAD DE LA DEMANDA DE TRAINING ES LA PALANCA DE MAYOR VALOR Y MENOR COSTE.

El Modelo 2, que permite desplazar libremente la carga de entrenamiento dentro del año, reduce el coste energético en 1.545.035 € anuales respecto al caso base (17,35 %), sin ningún CAPEX asociado. El Modelo 3, que restringe ese desplazamiento al interior de cada día, consigue un ahorro de 823.231 € anuales (9,25 %), igualmente con CAPEX nulo. Ambos resultados muestran que la simple reorganización temporal de las cargas flexibles ofrece un retorno inmediato, sin inversión en hardware ni riesgo financiero. En un centro de datos de esta escala, la gestión activa del perfil de demanda debería considerarse una medida de implantación prioritaria.

Una forma complementaria de interpretar estos ahorros es traducirlos a la cantidad de energía de training adicional que podría adquirirse cada año, al precio medio efectivamente pagado por el training en cada modelo:

Tabla 14. Training adicional comprable con el ahorro anual de cada modelo.

Modelo	Precio medio pagado por el training	Ahorro anual	Training extra comprable	% más training
M2	41,93 €/MWh	1.545.035 €	36.852.068 kWh/año	+49,5 %
M3	51,63 €/MWh	823.231 €	15.945.998 kWh/año	+21,4 %
M4	51,63 €/MWh	821.972 €	15.921.611 kWh/año	+21,4 %
M5	51,64 €/MWh	1.439.579 €	27.878.439 kWh/año	+37,5 %
M6	51,63 €/MWh	985.350 €	19.086.246 kWh/año	+25,6 %
M7	51,65 €/MWh	1.601.078 €	31.001.160 kWh/año	+41,7 %

Esta diferencia se aprecia también al desagregar la energía de training acumulada por periodo horario del día, para tres meses representativos de distintas épocas del año (enero, mayo y septiembre). El patrón horario es coherente con la lectura agregada: los Modelos 3, 4 y 6 mantienen perfiles próximos entre sí y al caso base a lo largo de las 24 horas, mientras que los Modelos 2 y 7 muestran una variabilidad horaria bastante mayor, con picos más

pronunciados en los meses de mayor volatilidad de precio, reflejo del margen adicional de desplazamiento del que disponen frente al resto de modelos.

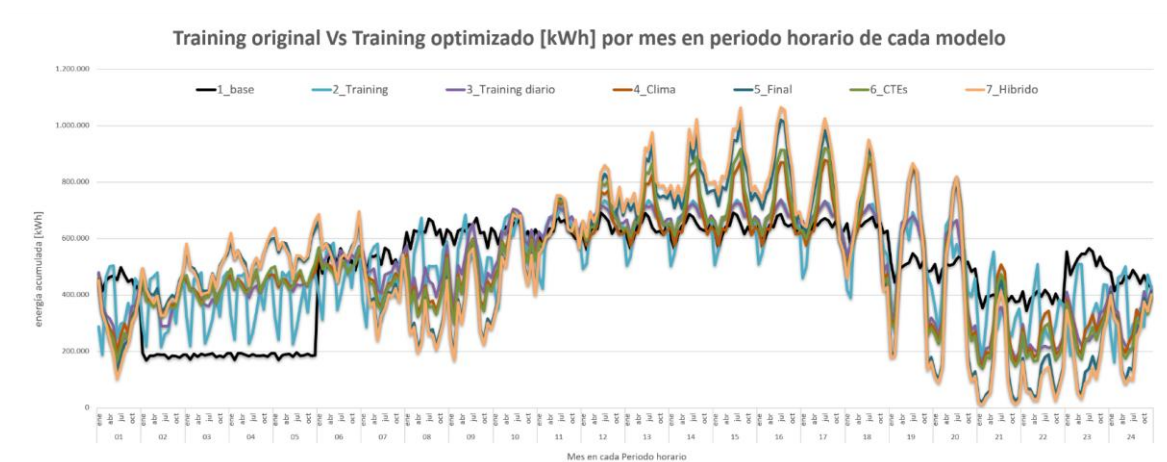


Figura 23. Training original vs. training optimizado (kWh) por mes en periodo horario de cada modelo.

7.1.2 SEGUNDA CONCLUSIÓN: LA REFRIGERACIÓN POR AMONIACO APORTA VALOR TÉCNICO PERO NO ECONÓMICO INCREMENTAL SOBRE M3.

El Modelo 4 incorpora un chiller de amoniaco (CAPEX 4,5 M€) y optimiza conjuntamente la refrigeración y el training diario. El coste anual resultante es prácticamente idéntico al de M3 (8.081.904 € frente a 8.080.644 €), con un ahorro marginal de apenas 1.259 € anuales. El VAN aparente de M4 (1.907.154 € a tasa del 8 %, TIR del 14,26 %) se apoya en el ahorro acumulado sobre el caso base, que procede casi íntegramente de la flexibilidad diaria introducida por M3 y que no requiere el chiller. Por ello, el Modelo 4 no se evalúa como una inversión autónoma: su finalidad es sustituir el perfil histórico de climatización por una representación dependiente de la temperatura y servir de base técnica para los modelos de almacenamiento. Frente a M3, no genera ahorro incremental suficiente para justificar por sí solo el CAPEX del sistema de refrigeración.

7.1.3 TERCERA CONCLUSIÓN: EL BESS SE ENCUENTRA EN EL UMBRAL DE VIABILIDAD Y SU RENTABILIDAD DEPENDE CRÍTICAMENTE DE LOS PARÁMETROS ADOPTADOS.

El Modelo 5 añade una batería Li-ion de 20 MWh, con un CAPEX total de 11,0 M€ incluyendo el chiller de M4 y los costes de integración. El coste anual baja a 7.464.296 €, con un ahorro bruto de 1.439.579 € anuales (16,17 % sobre M1). En el escenario base, sin crecimiento del precio de la energía y con una tasa de descuento del 8 %, el VAN es de -128.678 €, con una TIR del 7,82 %, prácticamente en el umbral del coste de capital. Por tanto, el BESS queda al borde de la viabilidad y su atractivo depende sobre todo de la evolución del precio eléctrico, del CAPEX de la batería y del posible aprovechamiento de la infraestructura de respaldo ya exigida por el nivel de disponibilidad del centro de datos. El análisis de sensibilidad cuantifica ese margen: basta un crecimiento del precio de la energía del 0,2 % anual, o una caída del CAPEX de la batería de apenas el 1,3 %, para que el VAN se vuelva positivo, de modo que el BESS queda muy cerca de la rentabilidad.

7.1.4 CUARTA CONCLUSIÓN: EL CTES ES LA ALTERNATIVA MÁS RENTABLE ENTRE LAS QUE REQUIEREN INVERSIÓN.

El Modelo 6 sustituye la batería por un tanque de almacenamiento térmico de frío (CTES) de 30.000 kWh_{th}, equivalente a 8.571 kWh_e con un COP de 3,5. El CAPEX total asciende a 7,4 M€ y no se considera reemplazo dentro del horizonte de 15 años. El coste anual es de 7.918.526 €, con un ahorro energético de 985.350 €/año. Al añadir el valor estimado de la capacidad IT liberada, 198.476 €/año, el beneficio bruto anual asciende a 1.183.826 €/año. Con estas hipótesis, el VAN del Modelo 6 es de 1.990.431 €, la TIR alcanza el 12,05 % y el payback simple es de 7 años. El CTES combina el CAPEX más bajo de las tres alternativas de almacenamiento y el mayor margen de seguridad frente a variaciones en el coste del tanque, la tasa de descuento y el beneficio anual, lo que lo convierte en la opción de menor riesgo entre las que requieren inversión.

7.1.5 QUINTA CONCLUSIÓN: EL HÍBRIDO BESS+CTES MAXIMIZA EL AHORRO ABSOLUTO CON UN MARGEN DE SEGURIDAD MÁS AJUSTADO, Y EL ORDEN DE INVERSIÓN IMPORTA.

El Modelo 7 demuestra que ambos almacenamientos pueden operar de forma coordinada y alcanzar el mayor ahorro absoluto de todos los modelos evaluados (1.601.078 €/año, un 17,98 %), con un CAPEX conjunto de 13.400.000 €, inferior a la suma de los proyectos individuales al compartir el chiller de amoníaco y los costes de integración. Sin embargo, su VAN (+422.196 €) y su TIR (8,49 %) quedan muy por debajo de los del CTES en solitario, porque ambos sistemas compiten por las mismas horas de precios bajos y el ahorro conjunto no es aditivo.

El análisis incremental muestra además que la secuencia de inversión importa: añadir el CTES sobre una batería existente es rentable (VAN incremental de +550.874 € y TIR del 11,5 %), mientras que añadir la batería sobre un CTES ya instalado destruye valor (VAN incremental de -1.568.235 € y TIR del 3,6 %). El híbrido queda así reservado para escenarios en los que el objetivo prioritario sea el ahorro absoluto y la disponibilidad de capital no sea la restricción principal, teniendo en cuenta que no conviene alcanzarlo partiendo del CTES: una vez instalado el tanque, la incorporación posterior de la batería no se justifica económicamente. En la práctica, dado que el BESS es una tecnología ya extendida en centros de datos, donde muchas instalaciones cuentan con baterías para respaldo eléctrico, el resultado sugiere que ampliar esas instalaciones existentes con un tanque CTES es una vía de inversión incremental de bajo riesgo, mientras que partir de un CTES para añadir después una batería no lo es.

7.1.6 SÍNTESIS GENERAL.

Tabla 15. Resumen de resultados y análisis de inversión de los siete modelos.

Modelo	Medida incorporada	Coste anual (€)	Ahorro (€/año)	Ahorro (%)	CAPEX (€)	VAN 8 %, 15 a (€)	TIR (%)
M1	Caso base	8.903.875	—	—	0	—	—
M2	Training anual	7.358.840	1.545.035	17,35 %	0	—	—
M3	Training diario	8.080.644	823.231	9,25 %	0	—	—
M4	+ Refrigeración amoníaco	8.081.904	821.972	9,23 %	4.500.000	n/a	n/a
M5	+ Batería BESS	7.464.296	1.439.579	16,17 %	11.000.000	-128.678	7,82 %
M6	+ CTES	7.918.526	1.183.826*	11,07 %**	7.400.000	1.990.431	12,05 %
M7	Híbrido BESS+CTES	7.302.797	1.799.489*	17,98 %**	13.400.000	422.196	8,49 %

*Beneficio bruto incluye valor de capacidad IT liberada. **% solo sobre ahorro energético puro.

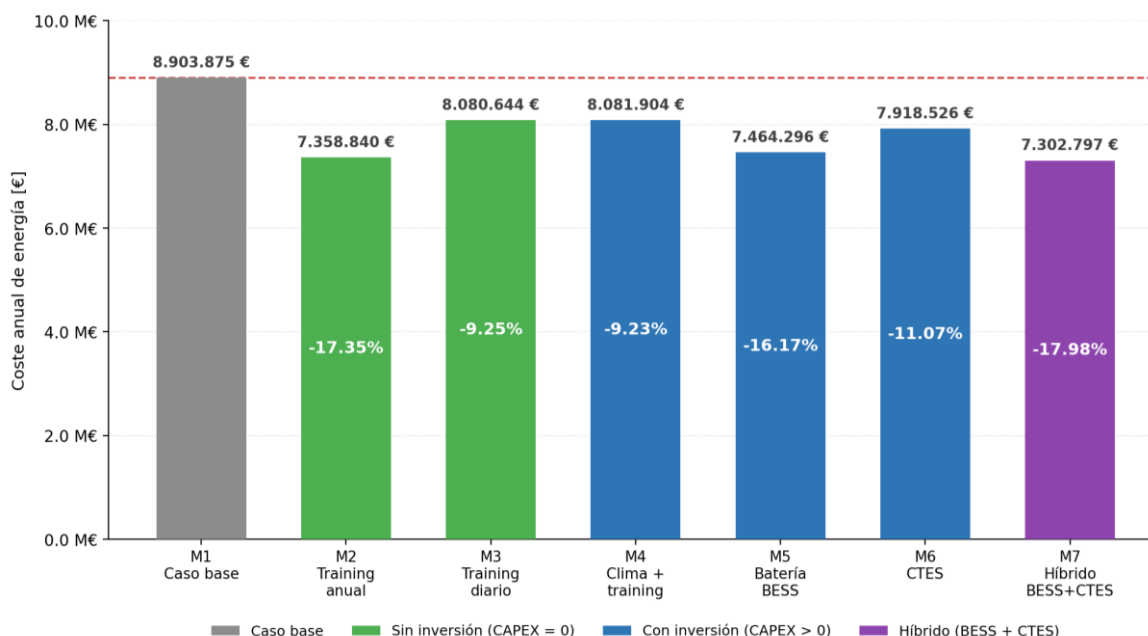


Figura 24. Evolución del beneficio acumulado de los modelos de inversión

Entre los modelos con inversión, M4, M6 y M7 son viables en el escenario base, mientras que M5 queda en el umbral. La recomendación principal es implantar la gestión activa de la demanda de training como primer paso, dado que el Modelo 3 genera ahorro sin CAPEX. A partir de ahí, M6 y M7 son las dos alternativas de inversión más sólidas del estudio, orientadas a criterios distintos: M6 ofrece la mejor rentabilidad por euro invertido y el mayor margen de seguridad frente a la incertidumbre; M7 ofrece el mayor ahorro absoluto, con un margen más ajustado que exige mayor confianza en las hipótesis del escenario base. La elección entre ambas depende de si el criterio del promotor prioriza minimizar el riesgo o maximizar el ahorro; en cualquier caso, el análisis incremental condiciona la ruta hacia M7: solo resulta rentable como ampliación de una instalación que ya dispone de batería, mientras que, instalado el CTES, incorporar después la batería no se justifica económicamente. El BESS en solitario queda cerca del umbral de viabilidad y podría resultar atractivo si bajan los costes de batería o aumenta la volatilidad del mercado eléctrico.

7.2 TRABAJOS FUTUROS

Los resultados de este trabajo abren varias líneas de investigación y desarrollo que quedan fuera del alcance del TFG pero que ampliarían su valor práctico:

7.2.1 PRECIOS INTRADIARIOS Y MERCADOS DE SERVICIOS DE AJUSTE.

Los modelos aquí desarrollados utilizan el precio OMIE del mercado diario como única señal de precio. Una extensión natural sería incorporar los mercados intradiarios y los servicios de ajuste del operador del sistema (REE), donde los precios pueden diferir significativamente del día anterior y abrir oportunidades adicionales de arbitraje.

7.2.2 HORIZONTE DE OPTIMIZACIÓN RODANTE.

El modelo actual optimiza sobre el año completo con información perfecta. Un planificador más realista utilizaría un horizonte rodante (rolling horizon) de 24-48 horas, con previsión de precios e incertidumbre explícita, lo que acercaría la formulación a las condiciones reales de operación.

7.2.3 INTEGRACIÓN DE ENERGÍA SOLAR FOTOVOLTAICA.

La adición de generación solar en cubierta o en parcela adyacente al CPD reduciría la dependencia de la red en las horas centrales del día y podría mejorar el perfil de carga del BESS o del CTES. Su modelización requiere datos de irradiación horaria y un módulo de gestión de excedentes.

7.2.4 OPTIMIZACIÓN MULTIOBJETIVO CON EMISIONES DE CO₂.

El criterio de optimización de este TFG es exclusivamente económico. Incorporar el coste de las emisiones de carbono, bien como restricción de huella máxima, bien como segundo objetivo, permitiría identificar la frontera de Pareto entre coste y emisiones, alineando el análisis con los marcos de sostenibilidad corporativa.

7.2.5 VALIDACIÓN CON DATOS REALES DE OPERACIÓN.

Los perfiles de demanda empleados son estimaciones basadas en literatura y en parámetros típicos del sector. Una validación con datos reales de un CPD en operación permitiría calibrar los modelos y cuantificar la diferencia entre el ahorro teórico y el alcanzable en la práctica.

7.2.6 ANÁLISIS DE DEGRADACIÓN DEL BESS CON MODELO DE ENVEJECIMIENTO.

El Modelo 5 supone un reemplazo de la batería en el año 10 como aproximación conservadora. Un modelo de degradación explícito, basado en ciclos de carga y temperatura, permitiría optimizar la estrategia de operación del BESS para maximizar su vida útil y mejorar la rentabilidad de la inversión.

Capítulo 8. BIBLIOGRAFÍA

- [1] International Energy Agency. «Energy and AI». IEA, París, abril de 2025. Disponible en:
<https://www.iea.org/reports/energy-and-ai>
- [2] Uptime Institute. «Uptime Institute Global Data Center Survey 2024». Uptime Intelligence, julio de 2024. Disponible en:
<https://uptimeinstitute.com/resources/research-and-reports>
- [3] Operador del Mercado Ibérico de Energía. «Detalle del funcionamiento del mercado intradiario y continuo». OMIE. Disponible en:
https://www.omie.es/sites/default/files/inline-files/mercados_intradiario_y_continuo.pdf
- [4] International Energy Agency. «Prices: Trends in wholesale markets differ across regions». Electricity Mid-Year Update 2025, IEA, París, julio de 2025. Disponible en:
<https://www.iea.org/reports/electricity-mid-year-update-2025/prices-trends-in-wholesale-markets-differ-across-regions>
- [5] NVIDIA. «NVIDIA 800 VDC Architecture Will Power the Next Generation of AI Factories». NVIDIA Technical Blog, 2025. Disponible en:
<https://developer.nvidia.com/blog/nvidia-800-v-hvdc-architecture-will-power-the-next-generation-of-ai-factories/>
- [6] ASHRAE. «ASHRAE Handbook — Refrigeration». American Society of Heating, Refrigerating and Air-Conditioning Engineers, Atlanta, GA, 2018.
- [7] International Energy Agency. «Batteries and Secure Energy Transitions». IEA, París, abril de 2024. Disponible en:
<https://www.iea.org/reports/batteries-and-secure-energy-transitions>
- [8] Cole, W. y Karmakar, A. «Cost Projections for Utility-Scale Battery Storage: 2023 Update». National Renewable Energy Laboratory, junio de 2023. Disponible en:
<https://www.nrel.gov/docs/fy23osti/85332.pdf>
- [9] Hart, W. E., Watson, J.-P. y Woodruff, D. L. «Pyomo: modeling and solving mathematical programs in Python». Mathematical Programming Computation, vol. 3, n.º 3, pp. 219-260, 2011.

(Véase también Bynum, M. L. et al., «Pyomo — Optimization Modeling in Python», 3.^a ed., Springer, 2021.)

[10] Huangfu, Q. y Hall, J. A. J. «Parallelizing the dual revised simplex method». *Mathematical Programming Computation*, vol. 10, n.º 1, pp. 119-142, 2018.

[11] Masanet, E., Shehabi, A., Lei, N., Smith, S. y Koomey, J. «Recalibrating global Data Center energy-use estimates». *Science*, vol. 367, n.º 6481, pp. 984-986, 2020.

[12] Qureshi, A., Weber, R., Balakrishnan, H., Gutttag, J. y Maggs, B. «Cutting the Electric Bill for Internet-Scale Systems». En *Proceedings of the ACM SIGCOMM 2009 Conference*, Barcelona, España, pp. 123-134, 2009.

[13] Radovanović, A., Koningstein, R., Schneider, I. et al. «Carbon-Aware Computing for Datacenters». *IEEE Transactions on Power Systems*, vol. 38, n.º 2, pp. 1270-1280, 2023.

[14] Lazard. «Levelized Cost of Energy+ (incluye Levelized Cost of Storage, versión 10.0)». Lazard, junio de 2025. Disponible en:

<https://www.lazard.com/research-insights/levelized-cost-of-energyplus/>

[15] Hertz, D. B. «Risk Analysis in Capital Investment». *Harvard Business Review*, vol. 42, n.º 1, pp. 95-106, 1964.

[16] Vertiv. «The Advantages of Using Lithium-ion Batteries as a Backup Power Source in Single-Phase UPS Applications for Remote and Edge Data Centers». Vertiv, White Paper, 2021.

[17] Brealey, R. A., Myers, S. C. y Allen, F. «Principles of Corporate Finance». 13.^a ed., McGraw-Hill Education, 2020.

[18] Operador del Mercado Ibérico de Energía (OMIE). «Resultados del mercado diario: precios horarios». OMIE, Madrid, 2025. Disponible en:

<https://www.omie.es>

[19] Open-Meteo. «Open-Meteo Weather API Documentation». Open-Meteo, 2024. Disponible en:

<https://open-meteo.com>

ANEXO I

Se recoge a continuación el código fuente en Python de los siete modelos de optimización desarrollados, implementados con la librería Pyomo y resueltos mediante el solver HiGHS.

8.1 *MODELO 1: CASO BASE SIN OPTIMIZACIÓN*

```
import pandas as pd
import pyomo.environ as pyo

# =====
# 1. LEER DATOS
# =====

RUTA_ARCHIVO = "datos_modelo.xlsx"

df = pd.read_excel(RUTA_ARCHIVO)

# Limpiar nombres de columnas
df.columns = df.columns.str.strip()

# Comprobar columnas necesarias
columnas_necesarias = [
    "hora",
    "datetime",
    "temp_ext",
    "carga_training_kwh",
    "carga_inferencia_kwh",
    "climatizacion_kwh",
    "comunes_kwh",
    "dc_kw",
    "dc_mwh",
    "precio_eur_mwh"
]

for col in columnas_necesarias:
```

```
if col not in df.columns:
    raise ValueError(f"Falta la columna: {col}")

# Ordenar por hora
df = df.sort_values("hora").reset_index(drop=True)

print("Filas leídas:", len(df))

# =====
# 2. PREPARAR PARÁMETROS
# =====

T = list(df["hora"].astype(int))

precio_eur_mwh = dict(zip(T, df["precio_eur_mwh"]))
temp_ext = dict(zip(T, df["temp_ext"]))
dc_kw = dict(zip(T, df["dc_kw"]))
datetime_dict = dict(zip(T, df["datetime"]))

# Convertir precio a €/kWh
precio_eur_kwh = {t: precio_eur_mwh[t] / 1000 for t in T}

# =====
# 3. CREAR MODELO
# =====

model = pyo.ConcreteModel()

model.T = pyo.Set(initialize=T)

model.precio = pyo.Param(model.T, initialize=precio_eur_kwh)
model.demanda = pyo.Param(model.T, initialize=dc_kw)

# =====
# 4. VARIABLES
# =====

model.energia_red = pyo.Var(model.T, domain=pyo.NonNegativeReals)

model.coste_total = pyo.Var(domain=pyo.NonNegativeReals)
```

```
# =====
# 5. RESTRICCIONES
# =====

def balance(model, t):
    return model.energia_red[t] == model.demanda[t]

model.balance = pyo.Constraint(model.T, rule=balance)

def coste(model):
    return model.coste_total == sum(model.energia_red[t] * model.precio[t]
    for t in model.T)

model.def_coste = pyo.Constraint(rule=coste)

# =====
# 6. FUNCIÓN OBJETIVO
# =====

model.obj = pyo.Objective(expr=model.coste_total, sense=pyo.minimize)

# =====
# 7. RESOLVER
# =====

solver = pyo.SolverFactory("appsi_highs")

result = solver.solve(model)

print("Estado:", result.solver.status)
print("Coste total anual (€):", pyo.value(model.coste_total))

# =====
# 8. EXPORTAR RESULTADOS
# =====

salida = pd.DataFrame({
    "hora": T,
    "datetime": [datetime_dict[t] for t in T],
    "energia_red_kwh": [pyo.value(model.energia_red[t]) for t in model.T],
    "precio_eur_kwh": [precio_eur_kwh[t] for t in T]
```

```
})

salida.to_excel("resultados_modelo_base.xlsx", index=False)

print("Archivo generado: resultados_modelo_base.xlsx")
```

8.2 *MODELO 2: FLEXIBILIDAD DE TRAINING ANUAL*

```
import pandas as pd
import pyomo.environ as pyo

# =====
# 1. LEER DATOS
# =====

RUTA_ARCHIVO = "datos_modelo.xlsx"

df = pd.read_excel(RUTA_ARCHIVO)
df.columns = df.columns.str.strip().str.replace(" ", "_", regex=False)

columnas_necesarias = [
    "hora",
    "datetime",
    "temp_ext",
    "carga_training_kwh",
    "carga_inferencia_kwh",
    "climatizacion_kwh",
    "comunes_kwh",
    "dc_kw",
    "dc_mwh",
    "precio_eur_mwh"
]

for col in columnas_necesarias:
    if col not in df.columns:
        raise ValueError(f"Falta la columna: {col}")
```

```
columnas_numericas = [
    "temp_ext",
    "carga_training_kwh",
    "carga_inferencia_kwh",
    "climatizacion_kwh",
    "comunes_kwh",
    "dc_kw",
    "dc_mwh",
    "precio_eur_mwh"
]

for col in columnas_numericas:
    df[col] = pd.to_numeric(df[col], errors="coerce")

df = df.sort_values("hora").reset_index(drop=True)

print("Filas leídas:", len(df))

# =====
# 2. PREPARAR PARÁMETROS
# =====

T = list(df["hora"].astype(int))

datetime_dict = dict(zip(T, df["datetime"]))
precio_eur_mwh = dict(zip(T, df["precio_eur_mwh"]))
precio_eur_kwh = {t: precio_eur_mwh[t] / 1000 for t in T}

training_base = dict(zip(T, df["carga_training_kwh"]))
inferencia = dict(zip(T, df["carga_inferencia_kwh"]))
climatizacion = dict(zip(T, df["climatizacion_kwh"]))
comunes = dict(zip(T, df["comunes_kwh"]))
temp_ext = dict(zip(T, df["temp_ext"]))

# Energía total anual de training que debe mantenerse
energia_total_training = df["carga_training_kwh"].sum()

# Límite máximo de training por hora
# Aquí lo ponemos como el máximo histórico observado
training_max_hora = df["carga_training_kwh"].max()
```

```
print("Energía total anual de training (kWh):", energia_total_training)
print("Potencia máxima histórica de training por hora (kWh):",
training_max_hora)

# Coste del caso base para comparar
coste_base = (
    (df["dc_kw"] * (df["precio_eur_mwh"] / 1000)).sum()
)
print("Coste caso base (€):", round(coste_base, 2))

# =====
# 3. CREAR MODELO
# =====

model = pyo.ConcreteModel()

model.T = pyo.Set(initialize=T, ordered=True)

model.precio = pyo.Param(model.T, initialize=precio_eur_kwh)
model.inferencia = pyo.Param(model.T, initialize=inferencia)
model.climatizacion = pyo.Param(model.T, initialize=climatizacion)
model.comunes = pyo.Param(model.T, initialize=comunes)

# =====
# 4. VARIABLES
# =====

# Training flexible
model.training_opt = pyo.Var(model.T, domain=pyo.NonNegativeReals)

# Demanda total del DC en cada hora
model.demanda_total = pyo.Var(model.T, domain=pyo.NonNegativeReals)

# Energía comprada de red
model.energia_red = pyo.Var(model.T, domain=pyo.NonNegativeReals)

# Coste total
model.coste_total = pyo.Var(domain=pyo.Reals)

# =====
```

5. RESTRICCIONES

=====

5.1 Demanda total = training flexible + resto de consumos

```
def demanda_total_rule(model, t):
```

```
    return model.demanda_total[t] == (  
        model.training_opt[t]  
        + model.inferencia[t]  
        + model.climatizacion[t]  
        + model.comunes[t]  
    )
```

```
model.demanda_total_def = pyo.Constraint(model.T, rule=demanda_total_rule)
```

5.2 La energía de red cubre exactamente la demanda

```
def balance_rule(model, t):
```

```
    return model.energia_red[t] == model.demanda_total[t]
```

```
model.balance = pyo.Constraint(model.T, rule=balance_rule)
```

5.3 Mantener la energía total anual de training

```
def energia_training_total_rule(model):
```

```
    return sum(model.training_opt[t] for t in model.T) ==  
    energia_total_training
```

```
model.energia_training_total =
```

```
pyo.Constraint(rule=energia_training_total_rule)
```

5.4 Límite máximo por hora de training

```
def training_max_rule(model, t):
```

```
    return model.training_opt[t] <= training_max_hora
```

```
model.training_max = pyo.Constraint(model.T, rule=training_max_rule)
```

5.5 Definición del coste total

```
def coste_total_rule(model):
```

```
    return model.coste_total == sum(  
        model.energia_red[t] * model.precio[t] for t in model.T  
    )
```

```
model.coste_total_def = pyo.Constraint(rule=coste_total_rule)
```

```
# =====
# 6. FUNCIÓN OBJETIVO
# =====

model.obj = pyo.Objective(expr=model.coste_total, sense=pyo.minimize)

# =====
# 7. RESOLVER
# =====

solver = pyo.SolverFactory("appsi_highs")
result = solver.solve(model)

print("Estado:", result.solver.status)
coste_opt = pyo.value(model.coste_total)
print("Coste optimizado (€):", round(coste_opt, 2))
print("Ahorro absoluto (€):", round(coste_base - coste_opt, 2))
print("Ahorro relativo (%):", round(100 * (coste_base - coste_opt) /
coste_base, 4))

# =====
# 8. EXPORTAR RESULTADOS
# =====

salida = pd.DataFrame({
    "hora": T,
    "datetime": [datetime_dict[t] for t in T],
    "temp_ext": [temp_ext[t] for t in T],
    "precio_eur_mwh": [precio_eur_mwh[t] for t in T],
    "training_original_kwh": [training_base[t] for t in T],
    "training_optimizado_kwh": [pyo.value(model.training_opt[t]) for t in
model.T],
    "inferencia_kwh": [inferencia[t] for t in T],
    "climatizacion_kwh": [climatizacion[t] for t in T],
    "comunes_kwh": [comunes[t] for t in T],
    "demanda_total_optimizada_kwh": [pyo.value(model.demanda_total[t]) for t
in model.T],
    "energia_red_kwh": [pyo.value(model.energia_red[t]) for t in model.T],
})
```

```
salida["coste_horario_original_eur"] = (  
    (salida["training_original_kwh"] + salida["inferencia_kwh"] +  
    salida["climatizacion_kwh"] + salida["comunes_kwh"])  
    * salida["precio_eur_mwh"] / 1000  
)  
  
salida["coste_horario_optimizado_eur"] = (  
    salida["demanda_total_optimizada_kwh"] * salida["precio_eur_mwh"] / 1000  
)  
  
salida.to_excel("resultados_modelo_training.xlsx", index=False)  
  
print("Archivo generado: resultados_modelo_training.xlsx")
```

8.3 *MODELO 3: FLEXIBILIDAD DE TRAINING DIARIA*

```
import pandas as pd  
import pyomo.environ as pyo  
  
# =====  
# 1. LEER DATOS  
# =====  
  
RUTA_ARCHIVO = "datos_modelo.xlsx"  
  
df = pd.read_excel(RUTA_ARCHIVO)  
df.columns = df.columns.str.strip().str.replace(" ", "_", regex=False)  
  
columnas_necesarias = [  
    "hora",  
    "datetime",  
    "temp_ext",  
    "carga_training_kwh",  
    "carga_inferencia_kwh",  
    "climatizacion_kwh",  
    "comunes_kwh",  
    "dc_kw",  
    "dc_mwh",  
    "precio_eur_mwh"  
]  
]
```

```
for col in columnas_necesarias:
    if col not in df.columns:
        raise ValueError(f"Falta la columna: {col}")

columnas_numericas = [
    "temp_ext",
    "carga_training_kwh",
    "carga_inferencia_kwh",
    "climatizacion_kwh",
    "comunes_kwh",
    "dc_kw",
    "dc_mwh",
    "precio_eur_mwh"
]

for col in columnas_numericas:
    df[col] = pd.to_numeric(df[col], errors="coerce")

df = df.sort_values("hora").reset_index(drop=True)

# Convertir datetime
df["datetime"] = pd.to_datetime(df["datetime"])

# Crear columna de día
df["dia"] = df["datetime"].dt.date

print("Filas leídas:", len(df))
print("Número de días:", df["dia"].nunique())

# =====
# 2. PREPARAR PARÁMETROS
# =====

T = list(df["hora"].astype(int))
D = sorted(df["dia"].unique())

datetime_dict = dict(zip(T, df["datetime"]))
dia_dict = dict(zip(T, df["dia"]))

precio_eur_mwh = dict(zip(T, df["precio_eur_mwh"]))
```

```
precio_eur_kwh = {t: precio_eur_mwh[t] / 1000 for t in T}

training_base = dict(zip(T, df["carga_training_kwh"]))
inferencia = dict(zip(T, df["carga_inferencia_kwh"]))
climatizacion = dict(zip(T, df["climatizacion_kwh"]))
comunes = dict(zip(T, df["comunes_kwh"]))
temp_ext = dict(zip(T, df["temp_ext"]))

# Energía diaria original de training
energia_training_diaria = (
    df.groupby("dia")["carga_training_kwh"].sum().to_dict()
)

# Máximo histórico horario de training
training_max_hora = df["carga_training_kwh"].max()

print("Potencia máxima histórica de training por hora (kWh):",
      training_max_hora)

# Coste del caso base
coste_base = (
    df["dc_kw"] * (df["precio_eur_mwh"] / 1000)
).sum()

print("Coste caso base (€):", round(coste_base, 2))

# Diccionario horas por día
horas_por_dia = {
    d: list(df.loc[df["dia"] == d, "hora"].astype(int))
    for d in D
}

# =====
# 3. CREAR MODELO
# =====

model = pyo.ConcreteModel()

model.T = pyo.Set(initialize=T, ordered=True)
model.D = pyo.Set(initialize=D, ordered=True)
```

```
model.precio = pyo.Param(model.T, initialize=precio_eur_kwh)
model.inferencia = pyo.Param(model.T, initialize=inferencia)
model.climatizacion = pyo.Param(model.T, initialize=climatizacion)
model.comunes = pyo.Param(model.T, initialize=comunes)

# =====
# 4. VARIABLES
# =====

model.training_opt = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.demanda_total = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.energia_red = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.coste_total = pyo.Var(domain=pyo.Reals)

# =====
# 5. RESTRICCIONES
# =====

# Demanda total
def demanda_total_rule(model, t):
    return model.demanda_total[t] == (
        model.training_opt[t]
        + model.inferencia[t]
        + model.climatizacion[t]
        + model.comunes[t]
    )

model.demanda_total_def = pyo.Constraint(model.T, rule=demanda_total_rule)

# Balance de energía
def balance_rule(model, t):
    return model.energia_red[t] == model.demanda_total[t]

model.balance = pyo.Constraint(model.T, rule=balance_rule)

# Mantener la energía diaria de training
def energia_training_diaria_rule(model, d):
    return sum(model.training_opt[t] for t in horas_por_dia[d]) ==
energia_training_diaria[d]
```

```
model.energia_training_diaria = pyo.Constraint(model.D,
rule=energia_training_diaria_rule)

# Límite máximo horario de training
def training_max_rule(model, t):
    return model.training_opt[t] <= training_max_hora

model.training_max = pyo.Constraint(model.T, rule=training_max_rule)

# Coste total
def coste_total_rule(model):
    return model.coste_total == sum(
        model.energia_red[t] * model.precio[t] for t in model.T
    )

model.coste_total_def = pyo.Constraint(rule=coste_total_rule)

# =====
# 6. FUNCIÓN OBJETIVO
# =====

model.obj = pyo.Objective(expr=model.coste_total, sense=pyo.minimize)

# =====
# 7. RESOLVER
# =====

solver = pyo.SolverFactory("appsi_highs")
result = solver.solve(model)

print("Estado:", result.solver.status)

coste_opt = pyo.value(model.coste_total)

print("Coste optimizado diario (€):", round(coste_opt, 2))
print("Ahorro absoluto (€):", round(coste_base - coste_opt, 2))
print("Ahorro relativo (%):", round(100 * (coste_base - coste_opt) /
coste_base, 4))

# =====
# 8. EXPORTAR RESULTADOS
```

```
# =====
```

```
salida = pd.DataFrame({
    "hora": T,
    "datetime": [datetime_dict[t] for t in T],
    "dia": [dia_dict[t] for t in T],
    "temp_ext": [temp_ext[t] for t in T],
    "precio_eur_mwh": [precio_eur_mwh[t] for t in T],
    "training_original_kwh": [training_base[t] for t in T],
    "training_optimizado_kwh": [pyo.value(model.training_opt[t]) for t in
model.T],
    "inferencia_kwh": [inferencia[t] for t in T],
    "climatizacion_kwh": [climatizacion[t] for t in T],
    "comunes_kwh": [comunes[t] for t in T],
    "demanda_total_optimizada_kwh": [pyo.value(model.demanda_total[t]) for t
in model.T],
    "energia_red_kwh": [pyo.value(model.energia_red[t]) for t in model.T],
})

salida["coste_horario_original_eur"] = (
    (salida["training_original_kwh"]
    + salida["inferencia_kwh"]
    + salida["climatizacion_kwh"]
    + salida["comunes_kwh"])
    * salida["precio_eur_mwh"] / 1000
)

salida["coste_horario_optimizado_eur"] = (
    salida["demanda_total_optimizada_kwh"] * salida["precio_eur_mwh"] / 1000
)

salida.to_excel("resultados_modelo_training_diario.xlsx", index=False)

print("Archivo generado: resultados_modelo_training_diario.xlsx")
```

8.4 *MODELO 4: MODELO CLIMÁTICO Y REFRIGERACIÓN POR AMONIACO*

```
import pandas as pd
import pyomo.environ as pyo

# =====
# 1. LEER DATOS
# =====

RUTA_ARCHIVO = "datos_modelo.xlsx"

df = pd.read_excel(RUTA_ARCHIVO)
df.columns = df.columns.str.strip().str.replace(" ", "_", regex=False)

columnas_necesarias = [
    "hora",
    "datetime",
    "temp_ext",
    "carga_training_kwh",
    "carga_inferencia_kwh",
    "climatizacion_kwh",
    "comunes_kwh",
    "dc_kw",
    "dc_mwh",
    "precio_eur_mwh"
]

for col in columnas_necesarias:
    if col not in df.columns:
        raise ValueError(f"Falta la columna: {col}")

columnas_numericas = [
    "temp_ext",
    "carga_training_kwh",
    "carga_inferencia_kwh",
    "climatizacion_kwh",
    "comunes_kwh",
    "dc_kw",

```

```
"dc_mwh",
"precio_eur_mwh"
]

for col in columnas_numericas:
    df[col] = pd.to_numeric(df[col], errors="coerce")

df = df.sort_values("hora").reset_index(drop=True)
df["datetime"] = pd.to_datetime(df["datetime"])
df["dia"] = df["datetime"].dt.date

print("Filas leídas:", len(df))
print("Número de días:", df["dia"].nunique())

# =====
# 2. PARÁMETROS Y CALIBRACIÓN
# =====

T = list(df["hora"].astype(int))
D = sorted(df["dia"].unique())

datetime_dict = dict(zip(T, df["datetime"]))
dia_dict = dict(zip(T, df["dia"]))

precio_eur_mwh = dict(zip(T, df["precio_eur_mwh"]))
precio_eur_kwh = {t: precio_eur_mwh[t] / 1000 for t in T}

training_base = dict(zip(T, df["carga_training_kwh"]))
inferencia = dict(zip(T, df["carga_inferencia_kwh"]))
comunes = dict(zip(T, df["comunes_kwh"]))
temp_ext = dict(zip(T, df["temp_ext"]))
clima_real = dict(zip(T, df["climatizacion_kwh"]))

# -----
# Modelo climático sencillo
# -----
T_REF = 20.0

df["exceso_temp"] = (df["temp_ext"] - T_REF).clip(lower=0)

clima_base = df["climatizacion_kwh"].min()
```

```
media_clima_real = df["climatizacion_kwh"].mean()
media_exceso_temp = df["exceso_temp"].mean()

if media_exceso_temp > 0:
    alpha = max(0, (media_clima_real - clima_base) / media_exceso_temp)
else:
    alpha = 0

print("Temperatura de referencia (°C):", T_REF)
print("Climatización base estimada (kWh):", round(clima_base, 4))
print("Alpha estimado (kWh/°C):", round(alpha, 4))

exceso_temp = dict(zip(T, df["exceso_temp"]))

# Energía diaria original de training
energia_training_diaria =
df.groupby("dia")["carga_training_kwh"].sum().to_dict()

# Máximo histórico horario de training
training_max_hora = df["carga_training_kwh"].max()

print("Potencia máxima histórica de training por hora (kWh):",
training_max_hora)

# Coste caso base real
coste_base = (df["dc_kw"] * (df["precio_eur_mwh"] / 1000)).sum()
print("Coste caso base (€):", round(coste_base, 2))

# Horas por día
horas_por_dia = {
    d: list(df.loc[df["dia"] == d, "hora"].astype(int))
    for d in D
}

# =====
# 3. CREAR MODELO
# =====

model = pyo.ConcreteModel()
```

```
model.T = pyo.Set(initialize=T, ordered=True)
model.D = pyo.Set(initialize=D, ordered=True)

model.precio = pyo.Param(model.T, initialize=precio_eur_kwh)
model.inferencia = pyo.Param(model.T, initialize=inferencia)
model.comunes = pyo.Param(model.T, initialize=comunes)
model.exceso_temp = pyo.Param(model.T, initialize=exceso_temp)

# =====
# 4. VARIABLES
# =====

model.training_opt = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.climatizacion_opt = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.demanda_total = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.energia_red = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.coste_total = pyo.Var(domain=pyo.Reals)

# =====
# 5. RESTRICCIONES
# =====

# 5.1 Climatización dependiente de temperatura
def climatizacion_rule(model, t):
    return model.climatizacion_opt[t] == clima_base + alpha *
model.exceso_temp[t]

model.climatizacion_def = pyo.Constraint(model.T, rule=climatizacion_rule)

# 5.2 Demanda total
def demanda_total_rule(model, t):
    return model.demanda_total[t] == (
        model.training_opt[t]
        + model.inferencia[t]
        + model.comunes[t]
        + model.climatizacion_opt[t]
    )

model.demanda_total_def = pyo.Constraint(model.T, rule=demanda_total_rule)

# 5.3 Balance
```

```
def balance_rule(model, t):
    return model.energia_red[t] == model.demanda_total[t]

model.balance = pyo.Constraint(model.T, rule=balance_rule)

# 5.4 Energía diaria de training
def energia_training_diaria_rule(model, d):
    return sum(model.training_opt[t] for t in horas_por_dia[d]) ==
energia_training_diaria[d]

model.energia_training_diaria = pyo.Constraint(model.D,
rule=energia_training_diaria_rule)

# 5.5 Límite máximo horario de training
def training_max_rule(model, t):
    return model.training_opt[t] <= training_max_hora

model.training_max = pyo.Constraint(model.T, rule=training_max_rule)

# 5.6 Coste total
def coste_total_rule(model):
    return model.coste_total == sum(
        model.energia_red[t] * model.precio[t] for t in model.T
    )

model.coste_total_def = pyo.Constraint(rule=coste_total_rule)

# =====
# 6. FUNCIÓN OBJETIVO
# =====

model.obj = pyo.Objective(expr=model.coste_total, sense=pyo.minimize)

# =====
# 7. RESOLVER
# =====

solver = pyo.SolverFactory("appsi_highs")
result = solver.solve(model)

print("Estado:", result.solver.status)
```

```
coste_opt = pyo.value(model.coste_total)

print("Coste optimizado con clima (€):", round(coste_opt, 2))
print("Ahorro absoluto vs base (€):", round(coste_base - coste_opt, 2))
print("Ahorro relativo vs base (%):", round(100 * (coste_base - coste_opt) /
coste_base, 4))

# =====
# 8. EXPORTAR RESULTADOS
# =====

salida = pd.DataFrame({
    "hora": T,
    "datetime": [datetime_dict[t] for t in T],
    "dia": [dia_dict[t] for t in T],
    "temp_ext": [temp_ext[t] for t in T],
    "exceso_temp": [exceso_temp[t] for t in T],
    "precio_eur_mwh": [precio_eur_mwh[t] for t in T],
    "training_original_kwh": [training_base[t] for t in T],
    "training_optimizado_kwh": [pyo.value(model.training_opt[t]) for t in
model.T],
    "climatizacion_real_kwh": [clima_real[t] for t in T],
    "climatizacion_modelada_kwh": [pyo.value(model.climatizacion_opt[t]) for
t in model.T],
    "inferencia_kwh": [inferencia[t] for t in T],
    "comunes_kwh": [comunes[t] for t in T],
    "demanda_total_optimizada_kwh": [pyo.value(model.demanda_total[t]) for t
in model.T],
    "energia_red_kwh": [pyo.value(model.energia_red[t]) for t in model.T],
})

salida["coste_horario_optimizado_eur"] = (
    salida["demanda_total_optimizada_kwh"] * salida["precio_eur_mwh"] / 1000
)

salida.to_excel("resultados_modelo_4_clima.xlsx", index=False)

print("Archivo generado: resultados_modelo_4_clima.xlsx")
```

8.5 *MODELO 5: SISTEMA COMPLETO CON ALMACENAMIENTO EN BATERÍA*

MODELO 5 – BESS Li-ion (Battery Energy Storage System)

Sistema: chiller de amoníaco (M4) + batería de iones de litio de 20 MWh

Lógica física:

- La batería carga en horas de precio bajo (compra electricidad barata).
- Descarga en horas de precio alto, reduciendo la compra a red.
- El optimizador decide cuándo cargar y descargar para minimizar coste total.
- El chiller de amoníaco sigue activo con la misma calibración que M4.

Variables de decisión:

- `training_opt[t]` – carga de training flexible (kWh_e)
- `carga_bateria[t]` – potencia de carga de la batería (kWh_e)
- `descarga_bat[t]` – potencia de descarga de la batería (kWh_e)
- `soc[t]` – estado de carga de la batería (kWh_e)
- `energia_red[t]` – energía comprada a red (kWh_e)

Balance eléctrico:

```
energia_red[t] + descarga_bat[t] == demanda_total[t] + carga_bateria[t]
=> energia_red[t] = demanda_total[t] + carga_bateria[t] - descarga_bat[t]
"""
```

```
import pandas as pd
import pyomo.environ as pyo

# =====
# 1. LEER DATOS
# =====

RUTA_ARCHIVO = "datos_modelo.xlsx"

df = pd.read_excel(RUTA_ARCHIVO)
df.columns = df.columns.str.strip().str.replace(" ", "_", regex=False)

columnas_necesarias = [
```

```
"hora", "datetime", "temp_ext",
"carga_training_kwh", "carga_inferencia_kwh",
"climatizacion_kwh", "comunes_kwh",
"dc_kw", "dc_mwh", "precio_eur_mwh",
]
for col in columnas_necesarias:
    if col not in df.columns:
        raise ValueError(f"Falta la columna: {col}")

columnas_numericas = [
    "temp_ext", "carga_training_kwh", "carga_inferencia_kwh",
    "climatizacion_kwh", "comunes_kwh", "dc_kw", "dc_mwh", "precio_eur_mwh",
]
for col in columnas_numericas:
    df[col] = pd.to_numeric(df[col], errors="coerce")

df = df.sort_values("hora").reset_index(drop=True)
df["datetime"] = pd.to_datetime(df["datetime"])
df["dia"] = df["datetime"].dt.date

print("Filas leídas:", len(df))
print("Número de días:", df["dia"].nunique())

# =====
# 2. PARÁMETROS
# =====

T = list(df["hora"].astype(int))
D = sorted(df["dia"].unique())

datetime_dict = dict(zip(T, df["datetime"]))
dia_dict      = dict(zip(T, df["dia"]))

precio_eur_mwh = dict(zip(T, df["precio_eur_mwh"]))
precio_eur_kwh = {t: precio_eur_mwh[t] / 1000 for t in T}

training_base = dict(zip(T, df["carga_training_kwh"]))
inferencia    = dict(zip(T, df["carga_inferencia_kwh"]))
comunes       = dict(zip(T, df["comunes_kwh"]))
temp_ext_d    = dict(zip(T, df["temp_ext"]))
```

```
# --- Demanda eléctrica de refrigeración (misma calibración que M4) ---
T_REF    = 20.0    # °C
COP      = 3.5    # COP del chiller de amoníaco (kWh_th / kWh_e)

df["exceso_temp"]    = (df["temp_ext"] - T_REF).clip(lower=0)
demanda_elec_base    = df["climatizacion_kwh"].min()
media_clima          = df["climatizacion_kwh"].mean()
media_exceso         = df["exceso_temp"].mean()
alpha = max(0, (media_clima - demanda_elec_base) / media_exceso) if
media_exceso > 0 else 0

df["demanda_elec_frio"] = demanda_elec_base + alpha * df["exceso_temp"]

demanda_elec_frio = dict(zip(T, df["demanda_elec_frio"]))
exceso_temp_d     = dict(zip(T, df["exceso_temp"]))

print(f"\nT_ref={T_REF}°C | demanda_elec_base={demanda_elec_base:.0f}
kWh_e "
      f"| alpha={alpha:.3f} kWh_e/°C | COP={COP}")
print(f"Demanda elec. frio - min={min(demanda_elec_frio.values()):.0f} "
      f"max={max(demanda_elec_frio.values()):.0f} "
      f"media={sum(demanda_elec_frio.values())/len(T):.0f} kWh_e/h")

# --- Training diario ---
energia_training_diaria =
df.groupby("dia")["carga_training_kwh"].sum().to_dict()
training_max_hora       = df["carga_training_kwh"].max()

# -----
# Parámetros de la batería Li-ion (BESS)
# -----
BAT_CAPACIDAD    = 20_000.0    # kWh - capacidad total de la batería
BAT_CARGA_MAX    = 5_000.0    # kWh/h - potencia máxima de carga (C/4)
BAT_DESCARGA_MAX = 5_000.0    # kWh/h - potencia máxima de descarga (C/4)
ETA_CARGA        = 0.95       # eficiencia de carga (one-way)
ETA_DESCARGA     = 0.95       # eficiencia de descarga (one-way)
SOC_INICIAL      = 0.50 * BAT_CAPACIDAD    # 50 % => 10.000 kWh al inicio

print(f"\n--- BESS Li-ion ---")
print(f"Capacidad: {BAT_CAPACIDAD:>10,.0f} kWh")
print(f"Potencia carga max: {BAT_CARGA_MAX:>10,.0f} kWh/h")
```

```

print(f"Potencia descarga max:    {BAT_DESCARGA_MAX:>10,.0f} kWh/h")
print(f"Eficiencia carga:         {ETA_CARGA*100:.0f} %")
print(f"Eficiencia descarga:      {ETA_DESCARGA*100:.0f} %")
print(f"SOC inicial:              {SOC_INICIAL:>10,.0f} kWh (50 %)")

# Coste sin optimización
coste_base = (df["dc_kw"] * (df["precio_eur_mwh"] / 1000)).sum()
print(f"\nCoste caso base: {coste_base:,.2f} €")

horas_por_dia = {
    d: list(df.loc[df["dia"] == d, "hora"].astype(int)) for d in D
}
T_ordenado = sorted(T)

# =====
# 3. CREAR MODELO
# =====

model = pyo.ConcreteModel()

model.T = pyo.Set(initialize=T, ordered=True)
model.D = pyo.Set(initialize=D, ordered=True)

model.precio          = pyo.Param(model.T, initialize=precio_eur_kwh)
model.inferencia       = pyo.Param(model.T, initialize=inferencia)
model.comunes         = pyo.Param(model.T, initialize=comunes)
model.demanda_elec_frio = pyo.Param(model.T, initialize=demanda_elec_frio)

# =====
# 4. VARIABLES
# =====

model.training_opt    = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.carga_bateria   = pyo.Var(model.T, domain=pyo.NonNegativeReals) #
kWh_e cargados
model.descarga_bat     = pyo.Var(model.T, domain=pyo.NonNegativeReals) #
kWh_e descargados
model.soc              = pyo.Var(model.T, domain=pyo.NonNegativeReals) #
kWh_e estado de carga
model.energia_red      = pyo.Var(model.T, domain=pyo.NonNegativeReals) #
kWh_e comprados

```

```
model.z_carga = pyo.Var(model.T, domain=pyo.Binary) #
1=cargando, 0=descargando

# =====
# 5. RESTRICCIONES
# =====

# 5.1 Balance eléctrico
# red + descarga == training + inferencia + comunes + frio + carga
# => red = demanda_total + carga - descarga
def balance_elec_rule(model, t):
    return model.energia_red[t] == (
        model.training_opt[t]
        + model.inferencia[t]
        + model.comunes[t]
        + model.demanda_elec_frio[t]
        + model.carga_bateria[t]
        - model.descarga_bat[t]
    )

model.balance_elec = pyo.Constraint(model.T, rule=balance_elec_rule)

# 5.2 SOC de la batería
# soc[t] = soc[t-1] + eta_carga * carga[t] - (1/eta_descarga) *
# descarga[t]
# Sin autodescarga (Li-ion: < 0.01 %/h, despreciable)
def soc_rule(model, t):
    idx = T_ordenado.index(t)
    soc_prev = SOC_INICIAL if idx == 0 else model.soc[T_ordenado[idx - 1]]
    return model.soc[t] == (
        soc_prev
        + ETA_CARGA * model.carga_bateria[t]
        - (1.0 / ETA_DESCARGA) * model.descarga_bat[t]
    )

model.soc_def = pyo.Constraint(model.T, rule=soc_rule)

# 5.3 Capacidad máxima de la batería
def soc_max_rule(model, t):
    return model.soc[t] <= BAT_CAPACIDAD
```

```
model.soc_max = pyo.Constraint(model.T, rule=soc_max_rule)

# 5.4 Potencia máxima de carga (solo si z_carga=1)
def carga_max_rule(model, t):
    return model.carga_bateria[t] <= BAT_CARGA_MAX * model.z_carga[t]

model.carga_max = pyo.Constraint(model.T, rule=carga_max_rule)

# 5.5 Potencia máxima de descarga (solo si z_carga=0)
def descarga_max_rule(model, t):
    return model.descarga_bat[t] <= BAT_DESCARGA_MAX * (1 -
model.z_carga[t])

model.descarga_max = pyo.Constraint(model.T, rule=descarga_max_rule)

# 5.6 Training diario conservado
def energia_training_diaria_rule(model, d):
    return sum(model.training_opt[t] for t in horas_por_dia[d]) ==
energia_training_diaria[d]

model.energia_training_diaria = pyo.Constraint(model.D,
rule=energia_training_diaria_rule)

# 5.7 Límite máximo horario de training
def training_max_rule(model, t):
    return model.training_opt[t] <= training_max_hora

model.training_max = pyo.Constraint(model.T, rule=training_max_rule)

# 5.8 Condición cíclica: la batería termina el año en el mismo SOC que
empieza
# (igualdad estricta – la batería no acumula energía de forma neta)
def soc_final_rule(model):
    return model.soc[T_ordenado[-1]] == SOC_INICIAL

model.soc_final = pyo.Constraint(rule=soc_final_rule)

# =====
# 6. FUNCIÓN OBJETIVO
# =====
```

```

model.obj = pyo.Objective(
    expr=sum(model.energia_red[t] * model.precio[t] for t in model.T),
    sense=pyo.minimize,
)

# =====
# 7. RESOLVER
# =====

solver = pyo.SolverFactory("appsi_highs")
result = solver.solve(model)

print("\nEstado solver:", result.solver.status)

coste_opt = pyo.value(model.obj)
print(f"Coste optimizado: {coste_opt:>12,.2f} €")
print(f"Ahorro absoluto: {coste_base - coste_opt:>12,.2f} €")
print(f"Ahorro relativo: {100*(coste_base - coste_opt)/coste_base:>11.4f}
%")

# =====
# 8. EXPORTAR RESULTADOS
# =====

carga_b = [pyo.value(model.carga_bateria[t]) for t in T]
descarga_b = [pyo.value(model.descarga_bat[t]) for t in T]
soc_b = [pyo.value(model.soc[t]) for t in T]

# Estadísticas de operación
total_cargado = sum(carga_b)
total_descargado = sum(descarga_b)
max_soc = max(soc_b)
horas_carga = sum(1 for x in carga_b if x > 10)
horas_descarga = sum(1 for x in descarga_b if x > 10)
ciclos_equiv = total_descargado / ETA_DESCARGA / BAT_CAPACIDAD

print(f"\n--- Resumen batería (BESS) ---")
print(f"Total cargado: {total_cargado:>12,.0f} kWh/año")
print(f"Total descargado: {total_descargado:>12,.0f} kWh/año")
print(f"SOC maximo alcanzado: {max_soc:>12,.0f} kWh")
print(f"Horas carga activa: {horas_carga:>12,.0f} h/año")

```

```
print(f"Horas descarga activa: {horas_descarga:>12,.0f} h/año")
print(f"Ciclos equivalentes: {ciclos_equiv:>12,.1f} ciclos/año")

salida = pd.DataFrame({
    "hora": T,
    "datetime": [datetime_dict[t] for t in T],
    "dia": [dia_dict[t] for t in T],
    "temp_ext": [temp_ext_d[t] for t in T],
    "exceso_temp": [exceso_temp_d[t] for t in T],
    "precio_eur_mwh": [precio_eur_mwh[t] for t in T],
    "training_original_kwh": [training_base[t] for t in T],
    "training_optimizado_kwh": [pyo.value(model.training_opt[t]) for t
in T],
    "demanda_elec_frio_base_kwh": [demanda_elec_frio[t] for t in T],
    "carga_bateria_kwh": carga_b,
    "descarga_bateria_kwh": descarga_b,
    "soc_bateria_kwh": soc_b,
    "inferencia_kwh": [inferencia[t] for t in T],
    "comunes_kwh": [comunes[t] for t in T],
    "energia_red_kwh": [pyo.value(model.energia_red[t]) for t
in T],
})

salida["coste_horario_optimizado_eur"] = (
    salida["energia_red_kwh"] * salida["precio_eur_mwh"] / 1000
)

salida.to_excel("resultados_modelo_5_final.xlsx", index=False)
print("\nArchivo generado: resultados_modelo_5_final.xlsx")
```

8.6 *MODELO 6: SISTEMA CTES CON ALMACENAMIENTO TÉRMICO DE FRÍO*

MODELO 6 – CTES (Cold Thermal Energy Storage – Almacenamiento Térmico de Frío)

=====

===

Sistema: chiller de amoníaco (M4) + tanque de almacenamiento térmico de frío

Lógica física:

- El chiller corre más de lo necesario en horas baratas y almacena el frío excedente en el tanque (gasta más electricidad ahora).
- En horas caras o de máxima demanda IT, descarga el tanque y corre el chiller por debajo de la demanda (ahorra electricidad).
- El frío almacenado se pierde lentamente (pérdidas térmicas del tanque).
- Al descargar, la electricidad liberada puede destinarse a servidores IA.

Formulación: unidades eléctricas equivalentes en todo el modelo.

- carga_frio[t] – electricidad extra consumida para cargar el tanque (kWh_e)
- descarga_frio[t] – electricidad ahorrada al descargar el tanque (kWh_e)
- soc_frio[t] – nivel del tanque en kWh_e equivalentes

La conversión térmica se hace vía COP: 1 kWh_e -> COP kWh_th almacenados. Al usar kWh_e como unidad única el balance eléctrico queda limpio y no hay que propagar el COP por las restricciones.

Variables de decisión:

- training_opt[t] – carga de training flexible (kWh_e)
 - carga_frio[t] – electricidad extra para cargar tanque (kWh_e)
 - descarga_frio[t] – electricidad ahorrada al descargar tanque (kWh_e)
 - soc_frio[t] – nivel del tanque (kWh_e equivalente)
 - energia_red[t] – energía comprada a red (kWh_e)
- """

```
import pandas as pd
import pyomo.environ as pyo

# =====
# 1. LEER DATOS
# =====

RUTA_ARCHIVO = "datos_modelo.xlsx"

df = pd.read_excel(RUTA_ARCHIVO)
df.columns = df.columns.str.strip().str.replace(" ", "_", regex=False)

columnas_necesarias = [
```

```
"hora", "datetime", "temp_ext",
"carga_training_kwh", "carga_inferencia_kwh",
"climatizacion_kwh", "comunes_kwh",
"dc_kw", "dc_mwh", "precio_eur_mwh",
]
for col in columnas_necesarias:
    if col not in df.columns:
        raise ValueError(f"Falta la columna: {col}")

columnas_numericas = [
    "temp_ext", "carga_training_kwh", "carga_inferencia_kwh",
    "climatizacion_kwh", "comunes_kwh", "dc_kw", "dc_mwh", "precio_eur_mwh",
]
for col in columnas_numericas:
    df[col] = pd.to_numeric(df[col], errors="coerce")

df = df.sort_values("hora").reset_index(drop=True)
df["datetime"] = pd.to_datetime(df["datetime"])
df["dia"] = df["datetime"].dt.date

print("Filas leídas:", len(df))
print("Numero de dias:", df["dia"].nunique())

# =====
# 2. PARÁMETROS
# =====

T = list(df["hora"].astype(int))
D = sorted(df["dia"].unique())

datetime_dict = dict(zip(T, df["datetime"]))
dia_dict      = dict(zip(T, df["dia"]))

precio_eur_mwh = dict(zip(T, df["precio_eur_mwh"]))
precio_eur_kwh = {t: precio_eur_mwh[t] / 1000 for t in T}

training_base = dict(zip(T, df["carga_training_kwh"]))
inferencia    = dict(zip(T, df["carga_inferencia_kwh"]))
comunes       = dict(zip(T, df["comunes_kwh"]))
temp_ext_d    = dict(zip(T, df["temp_ext"]))
```

```
# --- Demanda eléctrica de refrigeración (misma calibración que M4) ---
T_REF    = 20.0    # °C
COP      = 3.5    # COP del chiller de amoniaco (kWh_th / kWh_e) – valor
conservador amoniaco industrial

df["exceso_temp"]    = (df["temp_ext"] - T_REF).clip(lower=0)
demanda_elec_base    = df["climatizacion_kwh"].min()
media_clima          = df["climatizacion_kwh"].mean()
media_exceso         = df["exceso_temp"].mean()
alpha = max(0, (media_clima - demanda_elec_base) / media_exceso) if
media_exceso > 0 else 0

df["demanda_elec_frio"] = demanda_elec_base + alpha * df["exceso_temp"]

demanda_elec_frio = dict(zip(T, df["demanda_elec_frio"]))
exceso_temp_d     = dict(zip(T, df["exceso_temp"]))

print(f"\nT_ref={T_REF}C | demanda_elec_base={demanda_elec_base:.0f}
kWh_e "
      f"| alpha={alpha:.3f} kWh_e/C | COP={COP}")
print(f"Demanda elec. frio -- min={min(demanda_elec_frio.values()):.0f} "
      f"max={max(demanda_elec_frio.values()):.0f} "
      f"media={sum(demanda_elec_frio.values())/len(T):.0f} kWh_e/h")

# --- Training diario ---
energia_training_diaria =
df.groupby("dia")["carga_training_kwh"].sum().to_dict()
training_max_hora       = df["carga_training_kwh"].max()

# -----
# Parámetros del almacenamiento térmico de frío (CTES)
# Unidades: kWh_e equivalentes (electricidad ahorrada/gastada)
# -----
#
# Capacidad del tanque en kWh_th:      30.000 kWh_th
# Capacidad en kWh_e equiv:           30.000 / 3.5 = 8.571 kWh_e
# (= electricidad que el chiller hubiera necesitado para enfriar esa
# cantidad)
#
# Potencia max. de carga en kWh_th:   5.000 kWh_th/h
# En kWh_e equiv:                     5.000 / 3.5 = 1.429 kWh_e/h
```

```

CAP_TH_KWH          = 30_000.0      # kWh_th -- capacidad del tanque
CARGA_MAX_TH        = 5_000.0      # kWh_th/h -- potencia max. de carga
DESCARGA_MAX_TH     = 5_000.0      # kWh_th/h -- potencia max. de descarga
PERDIDA_HORA        = 0.005        # 0.5 % perdida termica/hora

# Equivalentes en kWh_e
CAP_E               = CAP_TH_KWH / COP      # 8.571 kWh_e
CARGA_MAX_E         = CARGA_MAX_TH / COP    # 1.429 kWh_e/h
DESCARGA_MAX_E      = DESCARGA_MAX_TH / COP # 1.429 kWh_e/h
SOC_INICIAL_E       = 0.30 * CAP_E          # 30 % de capacidad al inicio

ETA_CARGA_E         = 0.95      # eficiencia de carga del tanque (perdidas
transferencia)
ETA_DESCARGA_E      = 0.95      # eficiencia de descarga

print(f"\n--- CTES (en kWh_e equivalentes) ---")
print(f"Capacidad tanque:      {CAP_E:>10,.0f} kWh_e   ({CAP_TH_KWH:,>10,.0f} kWh_th)")
print(f"Potencia carga max:     {CARGA_MAX_E:>10,.0f} kWh_e/h
({CARGA_MAX_TH:,>10,.0f} kWh_th/h)")
print(f"Potencia descarga max:  {DESCARGA_MAX_E:>10,.0f} kWh_e/h
({DESCARGA_MAX_TH:,>10,.0f} kWh_th/h)")
print(f"Perdida termica/hora:   {PERDIDA_HORA*100:.1f} %")
print(f"SOC inicial:           {SOC_INICIAL_E:>10,.0f} kWh_e   (30 %)")

# Coste sin optimizacion
coste_base = (df["dc_kw"] * (df["precio_eur_mwh"] / 1000)).sum()
print(f"\nCoste caso base: {coste_base:,>10,.2f} EUR")

horas_por_dia = {
    d: list(df.loc[df["dia"] == d, "hora"].astype(int)) for d in D
}
T_ordenado = sorted(T)

# =====
# 3. CREAR MODELO
# =====

model = pyo.ConcreteModel()

```

```

model.T = pyo.Set(initialize=T, ordered=True)
model.D = pyo.Set(initialize=D, ordered=True)

model.precio          = pyo.Param(model.T, initialize=precio_eur_kwh)
model.inferencia       = pyo.Param(model.T, initialize=inferencia)
model.comunes         = pyo.Param(model.T, initialize=comunes)
model.demanda_elec_frio = pyo.Param(model.T, initialize=demanda_elec_frio)

# =====
# 4. VARIABLES
# =====

model.training_opt = pyo.Var(model.T, domain=pyo.NonNegativeReals)
model.carga_frio   = pyo.Var(model.T, domain=pyo.NonNegativeReals) #
kWh_e
model.descarga_frio = pyo.Var(model.T, domain=pyo.NonNegativeReals) #
kWh_e
model.soc_frio      = pyo.Var(model.T, domain=pyo.NonNegativeReals) #
kWh_e equiv.
model.energia_red   = pyo.Var(model.T, domain=pyo.NonNegativeReals) #
kWh_e
model.z_carga        = pyo.Var(model.T, domain=pyo.Binary)          #
1=cargando tanque, 0=descargando

# =====
# 5. RESTRICCIONES
# =====

# 5.1 Balance electrico
# red = training + inferencia + comunes + demanda_frio_base + carga_tanque
# - descarga_tanque
def balance_elec_rule(model, t):
    return model.energia_red[t] == (
        model.training_opt[t]
        + model.inferencia[t]
        + model.comunes[t]
        + model.demanda_elec_frio[t]
        + model.carga_frio[t]
        - model.descarga_frio[t]
    )

```

```
model.balance_elec = pyo.Constraint(model.T, rule=balance_elec_rule)

# 5.2 El chiller no puede correr en negativo:
# demanda_frio_base + carga - descarga >= 0
# => descarga <= demanda_frio_base + carga
def no_frio_negativo_rule(model, t):
    return model.descarga_frio[t] <= model.demanda_elec_frio[t] +
model.carga_frio[t]

model.no_frio_negativo = pyo.Constraint(model.T, rule=no_frio_negativo_rule)

# 5.3 SOC del tanque (en kWh_e equivalentes)
# soc[t] = soc[t-1] * (1 - perdida) + eta_carga * carga[t] - (1/eta_desc)
# * descarga[t]
def soc_frio_rule(model, t):
    idx = T_ordenado.index(t)
    soc_prev = SOC_INICIAL_E if idx == 0 else model.soc_frio[T_ordenado[idx
- 1]]
    return model.soc_frio[t] == (
        soc_prev * (1 - PERDIDA_HORA)
        + ETA_CARGA_E * model.carga_frio[t]
        - (1 / ETA_DESCARGA_E) * model.descarga_frio[t]
    )

model.soc_frio_def = pyo.Constraint(model.T, rule=soc_frio_rule)

# 5.4 Capacidad maxima del tanque
def soc_max_rule(model, t):
    return model.soc_frio[t] <= CAP_E

model.soc_max = pyo.Constraint(model.T, rule=soc_max_rule)

# 5.5 Potencia maxima de carga (solo si z_carga=1)
def carga_max_rule(model, t):
    return model.carga_frio[t] <= CARGA_MAX_E * model.z_carga[t]

model.carga_max = pyo.Constraint(model.T, rule=carga_max_rule)

# 5.6 Potencia maxima de descarga (solo si z_carga=0)
def descarga_max_rule(model, t):
    return model.descarga_frio[t] <= DESCARGA_MAX_E * (1 - model.z_carga[t])
```

```
model.descarga_max = pyo.Constraint(model.T, rule=descarga_max_rule)

# 5.7 Training diario conservado
def energia_training_diaria_rule(model, d):
    return sum(model.training_opt[t] for t in horas_por_dia[d]) ==
energia_training_diaria[d]

model.energia_training_diaria = pyo.Constraint(model.D,
rule=energia_training_diaria_rule)

# 5.8 Limite maximo horario de training
def training_max_rule(model, t):
    return model.training_opt[t] <= training_max_hora

model.training_max = pyo.Constraint(model.T, rule=training_max_rule)

# 5.9 Condicion ciclica: el tanque termina el año con al menos el SOC
inicial
def soc_frio_final_rule(model):
    return model.soc_frio[T_ordenado[-1]] >= SOC_INICIAL_E

model.soc_frio_final = pyo.Constraint(rule=soc_frio_final_rule)

# =====
# 6. FUNCIÓN OBJETIVO
# =====

model.obj = pyo.Objective(
    expr=sum(model.energia_red[t] * model.precio[t] for t in model.T),
    sense=pyo.minimize,
)

# =====
# 7. RESOLVER
# =====

solver = pyo.SolverFactory("appsi_highs")
result = solver.solve(model)

print("\nEstado solver:", result.solver.status)
```

```

coste_opt = pyo.value(model.obj)
print(f"Coste optimizado: {coste_opt:>12,.2f} EUR")
print(f"Ahorro absoluto: {coste_base - coste_opt:>12,.2f} EUR")
print(f"Ahorro relativo: {100*(coste_base - coste_opt)/coste_base:>11.4f}
%")

# =====
# 8. EXPORTAR RESULTADOS
# =====

carga_e = [pyo.value(model.carga_frio[t]) for t in T]
descarga_e = [pyo.value(model.descarga_frio[t]) for t in T]
soc_e = [pyo.value(model.soc_frio[t]) for t in T]

# Conversion de kWh_e a kWh_th para la salida (informativa)
carga_th = [c * COP for c in carga_e]
descarga_th = [d * COP for d in descarga_e]
soc_th = [s * COP for s in soc_e]

# Electricidad base del chiller (sin CTES) y con CTES
elec_chiller_base = [demanda_elec_frio[t] for t in T]
elec_chiller_opt = [demanda_elec_frio[t] + carga_e[i] - descarga_e[i]
for i, t in enumerate(T)]

# Capacidad IT liberada en cada hora: electricidad que el chiller ahorra
# al descargar el tanque (disponible para servidores de IA)
capacidad_it_liberada_kw = [max(0.0, descarga_e[i] - carga_e[i]) for i in
range(len(T))]

salida = pd.DataFrame({
    "hora": T,
    "datetime": [datetime_dict[t] for t in T],
    "dia": [dia_dict[t] for t in T],
    "temp_ext": [temp_ext_d[t] for t in T],
    "exceso_temp": [exceso_temp_d[t] for t in T],
    "precio_eur_mwh": [precio_eur_mwh[t] for t in T],
    "training_original_kwh": [training_base[t] for t in T],
    "training_optimizado_kwh": [pyo.value(model.training_opt[t]) for t
in T],
    "demanda_elec_frio_base_kwh": elec_chiller_base,

```

```
"carga_frio_kwh_e":          carga_e,
"descarga_frio_kwh_e":      descarga_e,
"soc_frio_kwh_e":           soc_e,
"carga_frio_kwh_th":        carga_th,
"descarga_frio_kwh_th":     descarga_th,
"soc_frio_kwh_th":          soc_th,
"elec_chiller_optimizado_kwh": elec_chiller_opt,
"capacidad_it_liberada_kw": capacidad_it_liberada_kw,
"inferencia_kwh":           [inferencia[t] for t in T],
"comunes_kwh":              [comunes[t] for t in T],
"energia_red_kwh":          [pyo.value(model.energia_red[t]) for t
in T],
})

salida["coste_horario_optimizado_eur"] = (
    salida["energia_red_kwh"] * salida["precio_eur_mwh"] / 1000
)

# Resumen operacion CTES
total_cargado      = salida["carga_frio_kwh_th"].sum()
total_descargado   = salida["descarga_frio_kwh_th"].sum()
max_soc_th         = salida["soc_frio_kwh_th"].max()
horas_descarga     = (salida["descarga_frio_kwh_e"] > 10).sum()
media_it_lib       = salida["capacidad_it_liberada_kw"].mean()
energia_it_total   = salida["capacidad_it_liberada_kw"].sum()
ciclos_equiv       = total_cargado / CAP_TH_KWH

print(f"\n--- Resumen almacenamiento termico (CTES) ---")
print(f"Total frio cargado al tanque:      {total_cargado:>12,.0f}
kWh_th/ano")
print(f"Total frio descargado:              {total_descargado:>12,.0f}
kWh_th/ano")
print(f"SOC maximo alcanzado:               {max_soc_th:>12,.0f} kWh_th")
print(f"Horas con descarga activa:           {horas_descarga:>12,.0f} h/ano")
print(f"Capacidad IT media liberada:          {media_it_lib:>12,.1f} kW_e")
print(f"Energia IT total liberada:           {energia_it_total:>12,.0f}
kWh_e/ano")
print(f"Ciclos equivalentes:                 {ciclos_equiv:>12,.1f} ciclos/ano")

salida.to_excel("resultados_modelo_6_ctes.xlsx", index=False)
print("\nArchivo generado: resultados_modelo_6_ctes.xlsx")
```

8.7 MODELO 7: SISTEMA HÍBRIDO CON BATERÍA BESS Y ALMACENAMIENTO TÉRMICO CTES

El script `modelo_tfg7.py` combina las lógicas de los Modelos 5 y 6 sobre un balance eléctrico conjunto: coordina simultáneamente la batería BESS y el tanque de frío CTES junto con la programación flexible del training, aplicando de forma simultánea las restricciones de operación de ambos sistemas de almacenamiento.

MODELO 7 – HÍBRIDO: BESS Li-ion + CTES (tanque de frío)

=====

Sistema: chiller de amoníaco (M4) + batería Li-ion 20 MWh (M5)
+ tanque de almacenamiento térmico de frío 30 MWh_th (M6)

Lógica física:

- La batería arbitra precios: carga barato, descarga caro (como M5).
- El tanque de frío desplaza el consumo del chiller: sobre-enfría en horas baratas y descarga frío en horas caras, liberando además capacidad eléctrica para servidores IA (como M6).
- El optimizador coordina ambos almacenamientos y el scheduling del training.

Variables de decisión:

- `training_opt[t]` – carga de training flexible (kWh_e)
- `carga_bateria[t]` / `descarga_bat[t]` / `soc_bat[t]` – BESS (kWh_e)
- `carga_frio[t]` / `descarga_frio[t]` / `soc_frio[t]` – CTES (kWh_e equiv.)
- `z_bat[t]`, `z_frio[t]` – binarias de no simultaneidad de cada almacenamiento
- `energia_red[t]` – energía comprada a red (kWh_e)

Balance eléctrico:

```
red = training + inferencia + comunes + frio_base
      + carga_bat - descarga_bat + carga_frio - descarga_frio
"""
```

```
import pandas as pd
import pyomo.environ as pyo

# =====
# 1. LEER DATOS
# =====

RUTA_ARCHIVO = "datos_modelo.xlsx"

df = pd.read_excel(RUTA_ARCHIVO)
df.columns = df.columns.str.strip().str.replace(" ", "_", regex=False)

columnas_necesarias = [
    "hora", "datetime", "temp_ext",
    "carga_training_kwh", "carga_inferencia_kwh",
    "climatizacion_kwh", "comunes_kwh",
    "dc_kw", "dc_mwh", "precio_eur_mwh",
]
for col in columnas_necesarias:
    if col not in df.columns:
        raise ValueError(f"Falta la columna: {col}")

columnas_numericas = [
    "temp_ext", "carga_training_kwh", "carga_inferencia_kwh",
    "climatizacion_kwh", "comunes_kwh", "dc_kw", "dc_mwh", "precio_eur_mwh",
]
for col in columnas_numericas:
    df[col] = pd.to_numeric(df[col], errors="coerce")

df = df.sort_values("hora").reset_index(drop=True)
df["datetime"] = pd.to_datetime(df["datetime"])
df["dia"] = df["datetime"].dt.date

print("Filas leídas:", len(df))
print("Numero de dias:", df["dia"].nunique())

# =====
# 2. PARÁMETROS
# =====

T = list(df["hora"].astype(int))
```

```
D = sorted(df["dia"].unique())

datetime_dict = dict(zip(T, df["datetime"]))
dia_dict      = dict(zip(T, df["dia"]))

precio_eur_mwh = dict(zip(T, df["precio_eur_mwh"]))
precio_eur_kwh = {t: precio_eur_mwh[t] / 1000 for t in T}

training_base = dict(zip(T, df["carga_training_kwh"]))
inferencia    = dict(zip(T, df["carga_inferencia_kwh"]))
comunes       = dict(zip(T, df["comunes_kwh"]))
temp_ext_d    = dict(zip(T, df["temp_ext"]))

# --- Demanda eléctrica de refrigeración (misma calibración que M4) ---
T_REF  = 20.0  # °C
COP     = 3.5  # COP del chiller de amoníaco (kWh_th / kWh_e) – valor
              conservador amoníaco industrial

df["exceso_temp"] = (df["temp_ext"] - T_REF).clip(lower=0)
demanda_elec_base = df["climatizacion_kwh"].min()
media_clima       = df["climatizacion_kwh"].mean()
media_exceso      = df["exceso_temp"].mean()
alpha = max(0, (media_clima - demanda_elec_base) / media_exceso) if
media_exceso > 0 else 0

df["demanda_elec_frio"] = demanda_elec_base + alpha * df["exceso_temp"]

demanda_elec_frio = dict(zip(T, df["demanda_elec_frio"]))
exceso_temp_d     = dict(zip(T, df["exceso_temp"]))

print(f"\nT_ref={T_REF}C | demanda_elec_base={demanda_elec_base:.0f}
kWh_e "
      f"| alpha={alpha:.3f} kWh_e/C | COP={COP}")

# --- Training diario ---
energia_training_diaria =
df.groupby("dia")["carga_training_kwh"].sum().to_dict()
training_max_hora       = df["carga_training_kwh"].max()

# -----
# Parámetros BESS Li-ion (idénticos a M5)
```

```
# -----
BAT_CAPACIDAD      = 20_000.0    # kWh
BAT_CARGA_MAX      = 5_000.0     # kWh/h
BAT_DESCARGA_MAX   = 5_000.0     # kWh/h
ETA_CARGA_BAT      = 0.95
ETA_DESCARGA_BAT   = 0.95
SOC_INICIAL_BAT    = 0.50 * BAT_CAPACIDAD    # 10.000 kWh

# -----
# Parámetros CTES (idénticos a M6)
# -----
CAP_TH_KWH         = 30_000.0    # kWh_th
CARGA_MAX_TH       = 5_000.0     # kWh_th/h
DESCARGA_MAX_TH    = 5_000.0     # kWh_th/h
PERDIDA_HORA       = 0.005       # 0.5 %/h

CAP_E              = CAP_TH_KWH / COP      # 8.571 kWh_e
CARGA_MAX_E        = CARGA_MAX_TH / COP    # 1.429 kWh_e/h
DESCARGA_MAX_E     = DESCARGA_MAX_TH / COP # 1.429 kWh_e/h
SOC_INICIAL_E      = 0.30 * CAP_E         # 30 %

ETA_CARGA_FRIO     = 0.95
ETA_DESCARGA_FRIO  = 0.95

print(f"\n--- BESS Li-ion ---")
print(f"Capacidad: {BAT_CAPACIDAD:,.0f} kWh | P.max: {BAT_CARGA_MAX:,.0f} kWh/h | eta: 95%/95% | SOC0: 50%")
print(f"--- CTES ---")
print(f"Capacidad: {CAP_E:,.0f} kWh_e ({CAP_TH_KWH:,.0f} kWh_th) | P.max: {CARGA_MAX_E:,.0f} kWh_e/h | perdida: 0.5%/h | SOC0: 30%")

# Coste sin optimización
coste_base = (df["dc_kw"] * (df["precio_eur_mwh"] / 1000)).sum()
print(f"\nCoste caso base: {coste_base:,.2f} EUR")

horas_por_dia = {
    d: list(df.loc[df["dia"] == d, "hora"].astype(int)) for d in D
}
T_ordenado = sorted(T)

# =====
```

3. CREAR MODELO

=====

```
model = pyo.ConcreteModel()
```

```
model.T = pyo.Set(initialize=T, ordered=True)
```

```
model.D = pyo.Set(initialize=D, ordered=True)
```

```
model.precio = pyo.Param(model.T, initialize=precio_eur_kwh)
```

```
model.inferencia = pyo.Param(model.T, initialize=inferencia)
```

```
model.comunes = pyo.Param(model.T, initialize=comunes)
```

```
model.demanda_elec_frio = pyo.Param(model.T, initialize=demanda_elec_frio)
```

=====

4. VARIABLES

=====

```
model.training_opt = pyo.Var(model.T, domain=pyo.NonNegativeReals)
```

BESS

```
model.carga_bateria = pyo.Var(model.T, domain=pyo.NonNegativeReals)
```

```
model.descarga_bat = pyo.Var(model.T, domain=pyo.NonNegativeReals)
```

```
model.soc_bat = pyo.Var(model.T, domain=pyo.NonNegativeReals)
```

```
model.z_bat = pyo.Var(model.T, domain=pyo.Binary) # 1=cargando,  
0=descargando
```

CTES

```
model.carga_frio = pyo.Var(model.T, domain=pyo.NonNegativeReals)
```

```
model.descarga_frio = pyo.Var(model.T, domain=pyo.NonNegativeReals)
```

```
model.soc_frio = pyo.Var(model.T, domain=pyo.NonNegativeReals)
```

```
model.z_frio = pyo.Var(model.T, domain=pyo.Binary) # 1=cargando  
tanque, 0=descargando
```

```
model.energia_red = pyo.Var(model.T, domain=pyo.NonNegativeReals)
```

=====

5. RESTRICCIONES

=====

5.1 Balance eléctrico

```
def balance_elec_rule(model, t):
```

```
        return model.energia_red[t] == (
            model.training_opt[t]
            + model.inferencia[t]
            + model.comunes[t]
            + model.demanda_elec_frio[t]
            + model.carga_bateria[t] - model.descarga_bat[t]
            + model.carga_frio[t] - model.descarga_frio[t]
        )

model.balance_elec = pyo.Constraint(model.T, rule=balance_elec_rule)

# 5.2 El chiller no puede correr en negativo (CTES)
def no_frio_negativo_rule(model, t):
    return model.descarga_frio[t] <= model.demanda_elec_frio[t] +
model.carga_frio[t]

model.no_frio_negativo = pyo.Constraint(model.T, rule=no_frio_negativo_rule)

# --- BESS ---
# 5.3 SOC batería (sin autodescarga)
def soc_bat_rule(model, t):
    idx = T_ordenado.index(t)
    soc_prev = SOC_INICIAL_BAT if idx == 0 else model.soc_bat[T_ordenado[idx
- 1]]
    return model.soc_bat[t] == (
        soc_prev
        + ETA_CARGA_BAT * model.carga_bateria[t]
        - (1.0 / ETA_DESCARGA_BAT) * model.descarga_bat[t]
    )

model.soc_bat_def = pyo.Constraint(model.T, rule=soc_bat_rule)

# 5.4 Capacidad máxima batería
def soc_bat_max_rule(model, t):
    return model.soc_bat[t] <= BAT_CAPACIDAD

model.soc_bat_max = pyo.Constraint(model.T, rule=soc_bat_max_rule)

# 5.5 Potencia máxima de carga batería (solo si z_bat=1)
def carga_bat_max_rule(model, t):
    return model.carga_bateria[t] <= BAT_CARGA_MAX * model.z_bat[t]
```

```
model.carga_bat_max = pyo.Constraint(model.T, rule=carga_bat_max_rule)

# 5.6 Potencia máxima de descarga batería (solo si z_bat=0)
def descarga_bat_max_rule(model, t):
    return model.descarga_bat[t] <= BAT_DESCARGA_MAX * (1 - model.z_bat[t])

model.descarga_bat_max = pyo.Constraint(model.T, rule=descarga_bat_max_rule)

# 5.7 Condición cíclica batería (igualdad estricta)
def soc_bat_final_rule(model):
    return model.soc_bat[T_ordenado[-1]] == SOC_INICIAL_BAT

model.soc_bat_final = pyo.Constraint(rule=soc_bat_final_rule)

# --- CTES ---
# 5.8 SOC del tanque (con pérdida térmica)
def soc_frio_rule(model, t):
    idx = T_ordenado.index(t)
    soc_prev = SOC_INICIAL_E if idx == 0 else model.soc_frio[T_ordenado[idx
- 1]]
    return model.soc_frio[t] == (
        soc_prev * (1 - PERDIDA_HORA)
        + ETA_CARGA_FRIO * model.carga_frio[t]
        - (1 / ETA_DESCARGA_FRIO) * model.descarga_frio[t]
    )

model.soc_frio_def = pyo.Constraint(model.T, rule=soc_frio_rule)

# 5.9 Capacidad máxima del tanque
def soc_frio_max_rule(model, t):
    return model.soc_frio[t] <= CAP_E

model.soc_frio_max = pyo.Constraint(model.T, rule=soc_frio_max_rule)

# 5.10 Potencia máxima de carga tanque (solo si z_frio=1)
def carga_frio_max_rule(model, t):
    return model.carga_frio[t] <= CARGA_MAX_E * model.z_frio[t]

model.carga_frio_max = pyo.Constraint(model.T, rule=carga_frio_max_rule)
```

```
# 5.11 Potencia máxima de descarga tanque (solo si z_frio=0)
def descarga_frio_max_rule(model, t):
    return model.descarga_frio[t] <= DESCARGA_MAX_E * (1 - model.z_frio[t])

model.descarga_frio_max = pyo.Constraint(model.T,
rule=descarga_frio_max_rule)

# 5.12 Condición cíclica tanque (al menos el SOC inicial)
def soc_frio_final_rule(model):
    return model.soc_frio[T_ordenado[-1]] >= SOC_INICIAL_E

model.soc_frio_final = pyo.Constraint(rule=soc_frio_final_rule)

# --- Training ---
# 5.13 Training diario conservado
def energia_training_diaria_rule(model, d):
    return sum(model.training_opt[t] for t in horas_por_dia[d]) ==
energia_training_diaria[d]

model.energia_training_diaria = pyo.Constraint(model.D,
rule=energia_training_diaria_rule)

# 5.14 Límite máximo horario de training
def training_max_rule(model, t):
    return model.training_opt[t] <= training_max_hora

model.training_max = pyo.Constraint(model.T, rule=training_max_rule)

# =====
# 6. FUNCIÓN OBJETIVO
# =====

model.obj = pyo.Objective(
    expr=sum(model.energia_red[t] * model.precio[t] for t in model.T),
    sense=pyo.minimize,
)

# =====
# 7. RESOLVER
# =====
```

```

solver = pyo.SolverFactory("appsi_highs")
result = solver.solve(model)

print("\nEstado solver:", result.solver.status)

coste_opt = pyo.value(model.obj)
print(f"Coste optimizado: {coste_opt:>12,.2f} EUR")
print(f"Ahorro absoluto: {coste_base - coste_opt:>12,.2f} EUR")
print(f"Ahorro relativo: {100*(coste_base - coste_opt)/coste_base:>11.4f}%")

# =====
# 8. EXPORTAR RESULTADOS
# =====

carga_b = [pyo.value(model.carga_bateria[t]) for t in T]
descarga_b = [pyo.value(model.descarga_bat[t]) for t in T]
soc_b = [pyo.value(model.soc_bat[t]) for t in T]

carga_f = [pyo.value(model.carga_frio[t]) for t in T]
descarga_f = [pyo.value(model.descarga_frio[t]) for t in T]
soc_f = [pyo.value(model.soc_frio[t]) for t in T]

# Conversión térmica CTES (informativa)
carga_f_th = [c * COP for c in carga_f]
descarga_f_th = [d * COP for d in descarga_f]
soc_f_th = [s * COP for s in soc_f]

# Capacidad IT liberada (solo por el CTES: chiller corre por debajo de la
demanda)
capacidad_it_liberada_kw = [max(0.0, descarga_f[i] - carga_f[i]) for i in
range(len(T))]

salida = pd.DataFrame({
    "hora": T,
    "datetime": [datetime_dict[t] for t in T],
    "dia": [dia_dict[t] for t in T],
    "temp_ext": [temp_ext_d[t] for t in T],
    "exceso_temp": [exceso_temp_d[t] for t in T],
    "precio_eur_mwh": [precio_eur_mwh[t] for t in T],
    "training_original_kwh": [training_base[t] for t in T],

```

```

        "training_optimizado_kwh":    [pyo.value(model.training_opt[t]) for t
in T],
        "demanda_elec_frio_base_kwh": [demanda_elec_frio[t] for t in T],
        "carga_bateria_kwh":         carga_b,
        "descarga_bateria_kwh":      descarga_b,
        "soc_bateria_kwh":           soc_b,
        "carga_frio_kwh_e":          carga_f,
        "descarga_frio_kwh_e":       descarga_f,
        "soc_frio_kwh_e":            soc_f,
        "carga_frio_kwh_th":         carga_f_th,
        "descarga_frio_kwh_th":      descarga_f_th,
        "soc_frio_kwh_th":           soc_f_th,
        "capacidad_it_liberada_kw":  capacidad_it_liberada_kw,
        "inferencia_kwh":            [inferencia[t] for t in T],
        "comunes_kwh":               [comunes[t] for t in T],
        "energia_red_kwh":           [pyo.value(model.energia_red[t]) for t
in T],
    })

salida["coste_horario_optimizado_eur"] = (
    salida["energia_red_kwh"] * salida["precio_eur_mwh"] / 1000
)

# Resumen operación
ciclos_bat = sum(descarga_b) / ETA_DESCARGA_BAT / BAT_CAPACIDAD
ciclos_ctes = sum(carga_f_th) / CAP_TH_KWH
energia_it = sum(capacidad_it_liberada_kw)

print(f"\n--- Resumen BESS ---")
print(f"Total cargado:      {sum(carga_b):>12,.0f} kWh/ano")
print(f"Total descargado:    {sum(descarga_b):>12,.0f} kWh/ano")
print(f"Ciclos equivalentes:{ciclos_bat:>12,.1f} ciclos/ano")
print(f"\n--- Resumen CTES ---")
print(f"Total frio cargado:    {sum(carga_f_th):>12,.0f} kWh_th/ano")
print(f"Total frio descargado: {sum(descarga_f_th):>12,.0f} kWh_th/ano")
print(f"Ciclos equivalentes:  {ciclos_ctes:>12,.1f} ciclos/ano")
print(f"Energia IT liberada:   {energia_it:>12,.0f} kWh_e/ano")

salida.to_excel("resultados_modelo_7_hibrido.xlsx", index=False)
print("\nArchivo generado: resultados_modelo_7_hibrido.xlsx")

```

8.8 ANÁLISIS ECONÓMICO

ANALISIS ECONOMICO – TFG Optimizacion Energetica Centros de Datos IA

=====

Version 5.0 – BESS (M5) vs CTES (M6) vs Hibrido (M7)

SISTEMAS MODELADOS:

- M1: Baseline (sin optimizacion)
- M2: Optimizacion training anual
- M3: Optimizacion training diario
- M4: Chiller amoniaco + training optimizado
- M5: M4 + bateria Li-ion 20 MWh (BESS) – arbitraje de precios electricos
- M6: M4 + tanque frio 30.000 kWh_th (CTES) – arbitraje termico + valor IT
- M7: M4 + BESS + CTES – hibrido con ambos almacenamientos

METRICAS CALCULADAS:

- VAN, TIR, Payback simple, Payback descontado (DPP), Indice rentabilidad (PI)
- LCOE desglosado por componente (EUR/MWh) – formula IEA/OCDE
- 4 tablas de sensibilidad (S1-S4) – enfocadas en M6
- Tornado chart (impacto de cada parametro en VAN M6)
- Breakeven analysis – umbrales de viabilidad M5 y M6
- Analisis complementario: VAN M6 solo ahorro energetico + VAN incremental

M6 vs M4

"""

```
import os
import warnings

import numpy as np
import numpy_financial as npf
import pandas as pd
from scipy import optimize

warnings.filterwarnings("ignore")

# =====
# 1. LEER RESULTADOS DE LOS 7 MODELOS
# =====
```

```
df_base = pd.read_excel("resultados_modelo_1_base.xlsx")
df_m2 = pd.read_excel("resultados_modelo_2_training.xlsx")
df_m3 = pd.read_excel("resultados_modelo_3_training_diario.xlsx")
df_m4 = pd.read_excel("resultados_modelo_4_clima.xlsx")
df_m5 = pd.read_excel("resultados_modelo_5_final.xlsx") # BESS
df_m6 = pd.read_excel("resultados_modelo_6_ctes.xlsx") # CTES
df_m7 = pd.read_excel("resultados_modelo_7_hibrido.xlsx") # BESS + CTES

coste_base = (df_base["energia_red_kwh"] * df_base["precio_eur_kwh"]).sum()
coste_m2 = df_m2["coste_horario_optimizado_eur"].sum()
coste_m3 = df_m3["coste_horario_optimizado_eur"].sum()
coste_m4 = df_m4["coste_horario_optimizado_eur"].sum()
coste_m5 = df_m5["coste_horario_optimizado_eur"].sum()
coste_m6 = df_m6["coste_horario_optimizado_eur"].sum()
coste_m7 = df_m7["coste_horario_optimizado_eur"].sum()

# Energia anual consumida (kWh) – denominador para LCOE
energia_anual_kwh = df_base["energia_red_kwh"].sum()

costes = {
    "M1 – Baseline": coste_base,
    "M2 – Training anual": coste_m2,
    "M3 – Training diario": coste_m3,
    "M4 – Clima + training": coste_m4,
    "M5 – BESS Li-ion 20MWh": coste_m5,
    "M6 – CTES tanque frio": coste_m6,
    "M7 – Hibrido BESS+CTES": coste_m7,
}

# =====
# 2. PARAMETROS ECONOMICOS
# =====

# --- Chiller de amoniaco (M4, M5, M6) ---
potencia_chiller_kw = 5_000 # kW nominal
coste_chiller_eur_kw = 900 # EUR/kW instalado
om_chiller_pct = 0.020 # 2.0 % CAPEX/año

# --- Bateria Li-ion (solo M5) ---
capacidad_bateria_kwh = 20_000 # kWh – del modelo BESS
coste_bateria_eur_kwh = 300 # EUR/kWh instalado (Li-ion 2024)
```

```

om_bateria_pct          = 0.020          # 2.0 % CAPEX/año

# --- Tanque de almacenamiento termico (solo M6) ---
capacidad_tanque_kwh_th = 30_000         # kWh_th – del modelo CTES
coste_tanque_eur_kwh_th = 80             # EUR/kWh_th instalado
om_tanque_pct           = 0.010         # 1.0 % CAPEX/año (mantenimiento
minimo)

# --- Integracion / SCADA / EPC ---
coste_integracion       = 500_000       # EUR (igual para M5 y M6)

# --- Horizonte ---
vida_util               = 15            # años de horizonte
tasa_descuento          = 0.08         # 8 % WACC
valor_residual_pct      = 0.10         # 10 % CAPEX al final del horizonte

escenarios_crecimiento = {
    "Sin crecimiento (0%)": 0.00,
    "Crecimiento moderado (2%)": 0.02,
    "Crecimiento alto (4%)": 0.04,
}
}

# =====
# 3. CAPEX Y OPEX
# =====

capex_chiller   = potencia_chiller_kw * coste_chiller_eur_kw #
4.500.000 EUR
capex_bateria   = capacidad_bateria_kwh * coste_bateria_eur_kwh #
6.000.000 EUR
capex_tanque    = capacidad_tanque_kwh_th * coste_tanque_eur_kwh_th #
2.400.000 EUR

capex_m4 = capex_chiller
capex_m5 = capex_chiller + capex_bateria + coste_integracion #
11.000.000 EUR
capex_m6 = capex_chiller + capex_tanque +
coste_integracion # 7.400.000 EUR
capex_m7 = capex_chiller + capex_bateria + capex_tanque +
coste_integracion # 13.400.000 EUR

```

```
opex_m4 = capex_chiller * om_chiller_pct
opex_m5 = capex_chiller * om_chiller_pct + capex_bateria * om_bateria_pct
opex_m6 = capex_chiller * om_chiller_pct + capex_tanque * om_tanque_pct
opex_m7 = (capex_chiller * om_chiller_pct + capex_bateria * om_bateria_pct
          + capex_tanque * om_tanque_pct)

capex_por_modelo = {
    "M1 - Baseline": 0,
    "M2 - Training anual": 0,
    "M3 - Training diario": 0,
    "M4 - Clima + training": capex_m4,
    "M5 - BESS Li-ion 20MWh": capex_m5,
    "M6 - CTES tanque frio": capex_m6,
    "M7 - Hibrido BESS+CTES": capex_m7,
}

opex_por_modelo = {
    "M1 - Baseline": 0,
    "M2 - Training anual": 0,
    "M3 - Training diario": 0,
    "M4 - Clima + training": opex_m4,
    "M5 - BESS Li-ion 20MWh": opex_m5,
    "M6 - CTES tanque frio": opex_m6,
    "M7 - Hibrido BESS+CTES": opex_m7,
}

# =====
# 4. VALOR DE CAPACIDAD IT (M6 y M7 - modelos con CTES)
# =====
# Al descargar frio del tanque, el chiller consume menos electricidad.
# Esa potencia liberada puede destinarse a servidores de IA.
# La BESS no genera este beneficio: su descarga aporta electricidad a la red
# interna pero no reduce el consumo del chiller.

precio_compute_eur_mwh = 120.0 # EUR/MWh_e de compute GPU (conservador)
factor_uso_capacidad = 0.60 # 60 % de la capacidad liberada -> ingresos

# Extraer de resultados M6
if "capacidad_it_liberada_kw" in df_m6.columns:
    energia_it_base_kwh = df_m6["capacidad_it_liberada_kw"].sum()
    potencia_it_media_kw = df_m6["capacidad_it_liberada_kw"].mean()
    horas_descarga_anual = (df_m6["descarga_frio_kwh_th"] > 10).sum()
```

```
energia_it_extra_kwh_ano = energia_it_base_kwh * factor_uso_capacidad
valor_capacidad_eur_ano = energia_it_extra_kwh_ano *
precio_compute_eur_mwh / 1000
else:
    potencia_it_media_kw = 800.0
    horas_descarga_anual = 2_500
    energia_it_base_kwh = potencia_it_media_kw * horas_descarga_anual
    energia_it_extra_kwh_ano = energia_it_base_kwh * factor_uso_capacidad
    valor_capacidad_eur_ano = energia_it_extra_kwh_ano *
precio_compute_eur_mwh / 1000
    print("AVISO: columna 'capacidad_it_liberada_kw' no encontrada. Usando
estimacion.")

# Valor capacidad IT de M7 (mismo mecanismo, con su propia serie horaria)
energia_it_m7_kwh = df_m7["capacidad_it_liberada_kw"].sum()
valor_capacidad_m7_ano = energia_it_m7_kwh * factor_uso_capacidad *
precio_compute_eur_mwh / 1000

# Beneficios anuales por modelo
ahorro_energia_m4 = coste_base - coste_m4
ahorro_energia_m5 = coste_base - coste_m5 # BESS: solo arbitraje
electrico
ahorro_energia_m6 = coste_base - coste_m6 # CTES: arbitraje termico
ahorro_energia_m7 = coste_base - coste_m7 # Hibrido
beneficio_total_m6 = ahorro_energia_m6 + valor_capacidad_eur_ano
beneficio_total_m7 = ahorro_energia_m7 + valor_capacidad_m7_ano

print(f"\n--- Valor capacidad IT (M6 - CTES) ---")
print(f"Energia total liberada (suma): {energia_it_base_kwh:>10,.0f}
kWh_e/ano")
print(f"Energia IT vendible
(x{factor_uso_capacidad:.0%}): {energia_it_extra_kwh_ano:>10,.0f}
kWh/ano")
print(f"Valor capacidad IT: {valor_capacidad_eur_ano:>10,.0f}
EUR/ano")
print(f"Ahorro energetico M6: {ahorro_energia_m6:>10,.0f}
EUR/ano")
print(f"Beneficio total M6: {beneficio_total_m6:>10,.0f}
EUR/ano")

print(f"\n--- Valor capacidad IT (M7 - Hibrido) ---")
```

```
print(f"Energia total liberada (suma):    {energia_it_m7_kwh:>10,.0f}  
kWh_e/ano")  
print(f"Valor capacidad IT:             {valor_capacidad_m7_ano:>10,.0f}  
EUR/ano")  
print(f"Ahorro energetico M7:           {ahorro_energia_m7:>10,.0f}  
EUR/ano")  
print(f"Beneficio total M7:             {beneficio_total_m7:>10,.0f}  
EUR/ano")  
  
# =====  
# 5. FUNCIONES DE CALCULO  
# =====  
  
def crf(r, n):  
    """Capital Recovery Factor."""  
    return r * (1 + r) ** n / ((1 + r) ** n - 1)  
  
def calcular_flujos(beneficio_bruto, capex, opex, crecimiento,  
                    r=None, n=None, vr_pct=None):  
    """  
    Flujos de caja desde año 0 (inversion) hasta n.  
    beneficio_bruto = ahorro energetico + valor capacidad IT (si aplica).  
    """  
    r      = tasa_descuento      if r      is None else r  
    n      = vida_util            if n      is None else n  
    vr_pct = valor_residual_pct if vr_pct is None else vr_pct  
  
    flujos = [-capex]  
    for year in range(1, n + 1):  
        factor_precio = (1 + crecimiento) ** (year - 1)  
        flujo = beneficio_bruto * factor_precio - opex  
        flujos.append(flujo)  
  
    flujos[-1] += capex * vr_pct  
    return flujos  
  
def _van(flujos, r=None):  
    r = tasa_descuento if r is None else r  
    return sum(cf / (1 + r) ** i for i, cf in enumerate(flujos))
```

```
def metricas(flujos, capex=0, r=None):
    """VAN, TIR, Payback simple, Payback descontado (DPP), PI."""
    r = tasa_descuento if r is None else r
    van = _van(flujos, r)

    try:
        tir = float(npf.irr(flujos))
        tir = tir if np.isfinite(tir) else None
    except Exception:
        tir = None

    acum, pb_simple = 0.0, np.inf
    for i, cf in enumerate(flujos):
        acum += cf
        if acum >= 0 and np.isinf(pb_simple):
            pb_simple = float(i)

    acum_desc, pb_desc = 0.0, np.inf
    for i, cf in enumerate(flujos):
        acum_desc += cf / (1 + r) ** i
        if acum_desc >= 0 and np.isinf(pb_desc):
            pb_desc = float(i)

    pi = (van + capex) / capex if capex > 0 else np.nan

    return {
        "van_eur": van,
        "tir": tir,
        "payback_anios": pb_simple,
        "dpp_anios": pb_desc,
        "pi": pi,
    }

def van_acumulado_df(flujos, r=None):
    r = tasa_descuento if r is None else r
    van_ac, rows = 0.0, []
    for i, cf in enumerate(flujos):
        cf_desc = cf / (1 + r) ** i
```

```

        van_ac += cf_desc
        rows.append({
            "anio": i,
            "flujo_eur": cf,
            "flujo_descontado_eur": cf_desc,
            "van_acumulado_eur": van_ac,
        })
    return pd.DataFrame(rows)

def desglose_lcoe(coste_anual_energia, capex, opex, energia_kwh_anual,
                  r=None, n=None, crecimiento=0.0):
    """LCOE desglosado (EUR/MWh) – formula IEA/OCDE."""
    r = tasa_descuento if r is None else r
    n = vida_util if n is None else n

    denominador = sum(energia_kwh_anual / (1 + r) ** t for t in range(1, n +
1))
    num_capital = capex
    num_om = sum(opex / (1 + r) ** t for t in range(1, n + 1))
    num_energia = sum(
        coste_anual_energia * (1 + crecimiento) ** (t - 1) / (1 + r) ** t
        for t in range(1, n + 1)
    )
    factor_mwh = 1000 / denominador

    return {
        "lcoe_capital_eur_mwh": round(num_capital * factor_mwh, 2),
        "lcoe_om_eur_mwh": round(num_om * factor_mwh, 2),
        "lcoe_energia_eur_mwh": round(num_energia * factor_mwh, 2),
        "lcoe_total_eur_mwh": round((num_capital + num_om + num_energia) *
factor_mwh, 2),
    }

# =====
# 6. COMPARACION ENTRE LOS 7 MODELOS
# =====

# Valor IT anual por modelo (0 para los que no tienen CTES)
valor_it_por_modelo = {nombre: 0.0 for nombre in costes}

```

```

valor_it_por_modelo["M6 - CTES tanque frio"] = valor_capacidad_eur_ano
valor_it_por_modelo["M7 - Hibrido BESS+CTES"] = valor_capacidad_m7_ano

filas_modelos = []
tablas_flujos = {}
filas_lcoe = []

for nombre, coste in costes.items():
    valor_it = valor_it_por_modelo[nombre]
    capex = capex_por_modelo[nombre]
    opex = opex_por_modelo[nombre]

    # LCOE
    d = desglose_lcoe(coste, capex, opex, energia_anual_kwh)
    d["modelo"] = nombre
    d["ahorro_energia_eur_mwh"] = round((coste_base - coste) /
(energia_anual_kwh / 1000), 2)
    d["valor_capacidad_it_eur_mwh"] = round(valor_it / (energia_anual_kwh /
1000), 2)
    filas_lcoe.append(d)

for nombre_crec, tasa_crec in escenarios_crecimiento.items():
    # Beneficio total: ahorro energetico + valor IT (M6 y M7)
    beneficio = (coste_base - coste) + valor_it
    flujos = calcular_flujos(beneficio, capex, opex, tasa_crec)
    m = metricas(flujos, capex=capex)

    filas_modelos.append({
        "modelo": nombre,
        "escenario_precio": nombre_crec,
        "coste_anual_eur": round(coste, 0),
        "ahorro_energia_eur": round(coste_base - coste, 0),
        "valor_it_eur": round(valor_it, 0),
        "beneficio_total_eur": round(beneficio, 0),
        "ahorro_pct_coste_base": round((coste_base - coste) /
coste_base * 100, 2),
        "capex_eur": capex,
        "opex_anual_eur": round(opex, 0),
        "van_eur": round(m["van_eur"], 0),
        "tir_pct": round(m["tir"] * 100, 2) if
m["tir"] else None,

```

```

        "payback_simple_anios":      m["payback_anios"] if
np.isfinite(m["payback_anios"]) else None,
        "payback_descontado_anios": m["dpp_anios"]      if
np.isfinite(m["dpp_anios"]) else None,
        "indice_rentabilidad_pi":   round(m["pi"], 3) if not
np.isnan(m["pi"]) else None,
        "viable":                   "SI" if m["van_eur"] > 0 else "NO",
    })

    tablas_flujos[f"{nombre} | {nombre_crec}"] =
van_acumulado_df(flujos)

df_modelos = pd.DataFrame(filas_modelos)
df_lcoe = pd.DataFrame(filas_lcoe)[[
    "modelo", "lcoe_capital_eur_mwh", "lcoe_om_eur_mwh",
    "lcoe_energia_eur_mwh", "lcoe_total_eur_mwh",
    "ahorro_energia_eur_mwh", "valor_capacidad_it_eur_mwh",
]]

# =====
# 7. FUNCIONES AUXILIARES PARA M6
# =====

ben_m6 = beneficio_total_m6

def van_m6(ben=None, cap=None, c=None, w=None, n=None, vr=None):
    b = ben_m6          if ben is None else ben
    ca = capex_m6       if cap is None else cap
    cr = 0.0            if c is None else c
    wa = tasa_descuento if w is None else w
    fl = calcular_flujos(b, ca, opex_m6, cr, r=wa, n=n, vr_pct=vr)
    return _van(fl, r=wa)

VAN_BASE_M6 = van_m6()

# Para M5
def van_m5(ben=None, cap=None, c=None, w=None):
    b = ahorro_energia_m5 if ben is None else ben
    ca = capex_m5         if cap is None else cap
    cr = 0.0              if c is None else c
    wa = tasa_descuento   if w is None else w

```

```
f1 = calcular_flujos(b, ca, opex_m5, cr, r=wa)
return _van(f1, r=wa)

VAN_BASE_M5 = van_m5()

# =====
# 8. TABLAS DE SENSIBILIDAD (enfocadas en M6)
# =====

# --- S1: Coste tanque (EUR/kWh_th) x Variacion beneficio total M6 ---
rango_coste_tanque = np.arange(40, 141, 20)      # 40 a 140 EUR/kWh_th
rango_var_beneficio = np.array([-0.20, -0.10, 0.00, 0.10, 0.20])

sens_s1 = pd.DataFrame(
    index = [f"{c} EUR/kWh_th" for c in rango_coste_tanque],
    columns = [f"Beneficio {int(v*100):+d}%" for v in rango_var_beneficio],
)
for ct in rango_coste_tanque:
    cap_s = capex_chiller + ct * capacidad_tanque_kwh_th + coste_integracion
    for vb in rango_var_beneficio:
        v = van_m6(ben=ben_m6 * (1 + vb), cap=cap_s)
        sens_s1.loc[f"{ct} EUR/kWh_th", f"Beneficio {int(vb*100):+d}%"] =
round(v, 0)
sens_s1 = sens_s1.astype(float)

# --- S2: WACC x Tasa crecimiento precio energia ---
rango_wacc = [0.04, 0.05, 0.06, 0.07, 0.08, 0.10, 0.12]
rango_crec = [0.00, 0.01, 0.02, 0.03, 0.04]

sens_s2 = pd.DataFrame(
    index = [f"WACC {int(w*100)}%" for w in rango_wacc],
    columns = [f"Crec.precio {int(c*100)}%" for c in rango_crec],
)
for w in rango_wacc:
    for c in rango_crec:
        v = van_m6(c=c, w=w)
        sens_s2.loc[f"WACC {int(w*100)}%", f"Crec.precio {int(c*100)}%"] =
round(v, 0)
sens_s2 = sens_s2.astype(float)

# --- S3: Precio compute (EUR/MWh) x Factor uso capacidad IT ---
```

```

rango_precio_compute = [60, 80, 100, 120, 150, 200]
rango_factor_uso      = [0.30, 0.40, 0.50, 0.60, 0.70, 0.80]

sens_s3 = pd.DataFrame(
    index    = [f"{p} EUR/MWh compute" for p in rango_precio_compute],
    columns  = [f"Factor uso {int(f*100)}%" for f in rango_factor_uso],
)
for p in rango_precio_compute:
    for f in rango_factor_uso:
        val_cap = energia_it_base_kwh * f * p / 1000
        ben_s    = ahorro_energia_m6 + val_cap
        v = van_m6(ben=ben_s)
        sens_s3.loc[f"{p} EUR/MWh compute", f"Factor uso {int(f*100)}%"] =
round(v, 0)
sens_s3 = sens_s3.astype(float)

# --- S4: Variacion beneficio total M6 x Variacion CAPEX M6 ---
rango_var_cap = np.array([-0.20, -0.10, 0.00, 0.10, 0.20])

sens_s4 = pd.DataFrame(
    index    = [f"Beneficio {int(v*100):+d}%" for v in rango_var_beneficio],
    columns  = [f"CAPEX {int(v*100):+d}%" for v in rango_var_cap],
)
for vb in rango_var_beneficio:
    for vc in rango_var_cap:
        v = van_m6(ben=ben_m6 * (1 + vb), cap=capex_m6 * (1 + vc))
        sens_s4.loc[f"Beneficio {int(vb*100):+d}%", f"CAPEX
{int(vc*100):+d}%"] = round(v, 0)
sens_s4 = sens_s4.astype(float)

# =====
# 9. TORNADO CHART – VAN M6
# =====

PARAMS_TORNADO = {
    "Coste tanque (EUR/kWh_th)": {"low": 40, "base": 80, "high":
140},
    "Beneficio total M6 (factor)": {"low": 0.80, "base": 1.00, "high":
1.20},
    "WACC": {"low": 0.05, "base": 0.08, "high":
0.12},

```

```

    "Precio compute GPU (EUR/MWh)": {"low": 60, "base": 120, "high":
200},
    "Factor uso capacidad IT": {"low": 0.30, "base": 0.60, "high":
0.80},
    "Crecimiento precio energia": {"low": 0.00, "base": 0.00, "high":
0.04},
    "Vida util proyecto (anos)": {"low": 10, "base": 15, "high":
20},
    "Valor residual (% CAPEX)": {"low": 0.00, "base": 0.10, "high":
0.20},
    "O&M total (% CAPEX/ano)": {"low": 0.012, "base": 0.017, "high":
0.025},
}

```

```

filas_tornado = []
for param, rng in PARAMS_TORNADO.items():
    resultados = {}
    for nivel in ("low", "high"):
        val = rng[nivel]

        if param == "Coste tanque (EUR/kWh_th)":
            cap_t = capex_chiller + val * capacidad_tanque_kwh_th +
coste_integracion
            van_t = van_m6(cap=cap_t)

        elif param == "Beneficio total M6 (factor)":
            van_t = van_m6(ben=ben_m6 * val)

        elif param == "WACC":
            van_t = van_m6(w=val)

        elif param == "Precio compute GPU (EUR/MWh)":
            val_cap = energia_it_base_kwh * factor_uso_capacidad * val /
1000
            van_t = van_m6(ben=ahorro_energia_m6 + val_cap)

        elif param == "Factor uso capacidad IT":
            val_cap = energia_it_base_kwh * val * precio_compute_eur_mwh /
1000
            van_t = van_m6(ben=ahorro_energia_m6 + val_cap)

```

```

elif param == "Crecimiento precio energia":
    van_t = van_m6(c=val)

elif param == "Vida util proyecto (anos)":
    van_t = van_m6(n=int(val))

elif param == "Valor residual (% CAPEX)":
    van_t = van_m6(vr=val)

elif param == "O&M total (% CAPEX/ano)":
    fl = calcular_flujos(ben_m6, capex_m6, capex_m6 * val, 0.0)
    van_t = _van(fl)

else:
    van_t = VAN_BASE_M6

resultados[nivel] = van_t

filas_tornado.append({
    "parametro":      param,
    "valor_low":      rng["low"],
    "valor_base":     rng["base"],
    "valor_high":     rng["high"],
    "van_low_eur":    round(resultados["low"], 0),
    "van_base_eur":   round(VAN_BASE_M6, 0),
    "van_high_eur":   round(resultados["high"], 0),
    "delta_low_eur":  round(resultados["low"] - VAN_BASE_M6, 0),
    "delta_high_eur": round(resultados["high"] - VAN_BASE_M6, 0),
    "rango_total_eur": round(abs(resultados["high"] -
resultados["low"])), 0),
    })

df_tornado = (
    pd.DataFrame(filas_tornado)
    .sort_values("rango_total_eur", ascending=False)
    .reset_index(drop=True)
)

# =====
# 10. BREAKEVEN – M5 y M6
# =====

```

```
def _buscar_raiz(f, lo, hi, xtol=1.0):
    try:
        fa, fb = f(lo), f(hi)
        if fa * fb > 0:
            return None
        return optimize.brentq(f, lo, hi, xtol=xtol, maxiter=300)
    except Exception:
        return None

# === Breakeven M5 (BESS) ===
ben_be_m5 = _buscar_raiz(lambda b: van_m5(ben=b), lo=0,
hi=ahorro_energia_m5 * 8, xtol=1.0)
cap_be_m5 = _buscar_raiz(lambda c: van_m5(cap=c), lo=0, hi=capex_m5 * 6,
xtol=1.0)
wacc_be_m5 = _buscar_raiz(lambda w: van_m5(w=w), lo=0.001, hi=0.40,
xtol=1e-5)

filas_be_m5 = [
    {
        " analisis": "M5 BESS – Beneficio minimo anual (VAN = 0)",
        " breakeven": f"{ben_be_m5:,.0f} EUR/ano" if ben_be_m5 else "No
cruza en rango",
        " actual": f"{ahorro_energia_m5:,.0f} EUR/ano",
        " margen": f"{(ahorro_energia_m5 -
ben_be_m5)/ben_be_m5*100:+.1f}%" if ben_be_m5 else "-",
    },
    {
        " analisis": "M5 BESS – CAPEX maximo admisible (VAN = 0)",
        " breakeven": f"{cap_be_m5:,.0f} EUR" if cap_be_m5 else "No
cruza en rango",
        " actual": f"{capex_m5:,.0f} EUR",
        " margen": f"{(cap_be_m5 - capex_m5)/capex_m5*100:+.1f}%" if
cap_be_m5 else "-",
    },
    {
        " analisis": "M5 BESS – WACC maximo = TIR (VAN = 0)",
        " breakeven": f"{wacc_be_m5*100:.1f}%" if wacc_be_m5 else "No
cruza en rango",
        " actual": f"WACC base: {tasa_descuento*100:.0f}%",
```

```

        "margen":      f"{{(wacc_be_m5 - tasa_descuento)*100:+.1f}} pp" if
wacc_be_m5 else "-",
    },
]

# === Breakeven M6 (CTES) ===
ben_be_m6 = _buscar_raiz(lambda b: van_m6(ben=b), lo=0, hi=ben_m6 * 6,
xtol=1.0)
cap_be_m6 = _buscar_raiz(lambda c: van_m6(cap=c), lo=0, hi=capex_m6 * 8,
xtol=1.0)
crec_be_m6 = _buscar_raiz(lambda c: van_m6(c=c), lo=0.0, hi=0.30, xtol=1e-
5)
wacc_be_m6 = _buscar_raiz(lambda w: van_m6(w=w), lo=0.001, hi=0.40,
xtol=1e-5)

def van_vs_precio_compute(p):
    val_cap = energia_it_base_kwh * factor_uso_capacidad * p / 1000
    return van_m6(ben=ahorro_energia_m6 + val_cap)

pcomp_be = _buscar_raiz(van_vs_precio_compute, lo=0.0, hi=500.0, xtol=0.01)

filas_be_m6 = [
    {
        " analisis": "M6 CTES – Beneficio minimo anual (VAN = 0)",
        " breakeven": f"{{ben_be_m6:,.0f}} EUR/ano" if ben_be_m6 else "No cruza
en rango",
        " actual": f"{{ben_m6:,.0f}} EUR/ano",
        " margen": f"{{(ben_m6 - ben_be_m6)/ben_be_m6*100:+.1f}}% sobre
umbral" if ben_be_m6 else "-",
    },
    {
        " analisis": "M6 CTES – CAPEX maximo admisible (VAN = 0)",
        " breakeven": f"{{cap_be_m6:,.0f}} EUR" if cap_be_m6 else "No cruza en
rango",
        " actual": f"{{capex_m6:,.0f}} EUR",
        " margen": f"{{(cap_be_m6 - capex_m6)/capex_m6*100:+.1f}}%" if
cap_be_m6 else "-",
    },
    {
        " analisis": "M6 CTES – Crecimiento minimo precio energia (VAN =
0)",

```

```

        "breakeven": f"{crec_be_m6*100:.2f}%/ano" if crec_be_m6 is not None
    else "Ya viable a 0%",
    "actual":      "Escenarios: 0%, 2%, 4%",
    "margen":      "-",
    },
    {
        "analisis": "M6 CTES – WACC maximo = TIR (VAN = 0)",
        "breakeven": f"{wacc_be_m6*100:.1f}%" if wacc_be_m6 else "No cruza
en rango",
        "actual":    f"WACC base: {tasa_descuento*100:.0f}%",
        "margen":    f"{(wacc_be_m6 - tasa_descuento)*100:+.1f} pp" if
wacc_be_m6 else "-",
    },
    {
        "analisis": "M6 CTES – Precio compute minimo para VAN = 0
(EUR/MWh)",
        "breakeven": f"{pcomp_be:.1f} EUR/MWh" if pcomp_be else "No
necesario (ahorro ya suficiente)",
        "actual":    f"{precio_compute_eur_mwh} EUR/MWh",
        "margen":    f"{precio_compute_eur_mwh - pcomp_be:.1f} EUR/MWh
margen" if pcomp_be else "-",
    },
]

df_breakeven = pd.DataFrame(filas_be_m5 + filas_be_m6)

# Curvas para graficas
_beneficios = np.linspace(ben_m6 * 0.20, ben_m6 * 2.5, 60)
_capexes    = np.linspace(capex_m6 * 0.20, capex_m6 * 2.5, 60)
_waccs      = np.linspace(0.02, 0.25, 60)
_crecs      = np.linspace(0.00, 0.12, 60)
_pcomputes  = np.linspace(0, 250, 60)

df_curva_ben = pd.DataFrame({"beneficio_eur":      _beneficios.round(0),
                             "van_m6_eur":        [round(van_m6(ben=b)
, 0) for b in _beneficios]})
df_curva_cap = pd.DataFrame({"capex_eur":          _capexes.round(0),
                             "van_m6_eur":        [round(van_m6(cap=c)
, 0) for c in _capexes]})
df_curva_wacc = pd.DataFrame({"wacc_pct":          (_waccs *
100).round(2),

```

```

                                "van_m5_eur":          [round(van_m5(w=w),
0) for w in _waccs],
                                "van_m6_eur":          [round(van_m6(w=w),
0) for w in _waccs]])
df_curva_crec = pd.DataFrame({"crec_precio_pct":      (_crecs *
100).round(2),
                                "van_m6_eur":          [round(van_m6(c=c),
0) for c in _crecs]])
df_curva_pcomp = pd.DataFrame({
    "precio_compute_eur_mwh": _pcomputes.round(1),
    "van_m6_eur": [
        round(van_m6(ben=ahorro_energia_m6 + energia_it_base_kwh *
factor_uso_capacidad * p / 1000), 0)
        for p in _pcomputes
    ],
})

# =====
# 11. ANALISIS COMPLEMENTARIO (M6)
#     A) VAN M6 solo ahorro energetico (sin valor IT)
#     B) VAN incremental M6 sobre M4 (solo el tanque)
# =====

# --- A) VAN M6 solo ahorro energetico ---
flujos_solo_ahorro = calcular_flujos(ahorro_energia_m6, capex_m6, opex_m6,
0.0)
metricas_solo_ahorro = metricas(flujos_solo_ahorro, capex=capex_m6)

# --- B) VAN incremental M6 vs M4 ---
beneficio_incr_m6 = (ahorro_energia_m6 - ahorro_energia_m4) +
valor_capacidad_eur_ano
capex_incremental_m6 = capex_tanque + coste_integracion          # 2.900.000
EUR
opex_incremental_m6 = capex_tanque * om_tanque_pct                # 24.000
EUR/año

flujos_incr = calcular_flujos(beneficio_incr_m6, capex_incremental_m6,
opex_incremental_m6, 0.0)
metricas_incr = metricas(flujos_incr, capex=capex_incremental_m6)

# --- C) VAN incremental M7 vs M6 (añadir la bateria al CTES) ---

```

```

beneficio_incr_m7m6 = (ahorro_energia_m7 - ahorro_energia_m6) +
(valor_capacidad_m7_ano - valor_capacidad_eur_ano)
capex_incr_m7m6      = capex_bateria                                # 6.000.000
EUR
opex_incr_m7m6       = capex_bateria * om_bateria_pct              # 120.000
EUR/año

flujos_m7m6 = calcular_flujos(beneficio_incr_m7m6, capex_incr_m7m6,
opex_incr_m7m6, 0.0)
metricas_m7m6 = metricas(flujos_m7m6, capex=capex_incr_m7m6)

# --- D) VAN incremental M7 vs M5 (añadir el tanque a la bateria) ---
beneficio_incr_m7m5 = (ahorro_energia_m7 - ahorro_energia_m5) +
valor_capacidad_m7_ano
capex_incr_m7m5      = capex_tanque                                # 2.400.000
EUR
opex_incr_m7m5       = capex_tanque * om_tanque_pct                # 24.000
EUR/año

flujos_m7m5 = calcular_flujos(beneficio_incr_m7m5, capex_incr_m7m5,
opex_incr_m7m5, 0.0)
metricas_m7m5 = metricas(flujos_m7m5, capex=capex_incr_m7m5)

df_analisis_compl = pd.DataFrame([
    {
        "analisis":      "M6 solo ahorro energetico (sin valor IT)",
        "capex_eur":      capex_m6,
        "beneficio_eur_ano": round(ahorro_energia_m6, 0),
        "van_eur":         round(metricas_solo_ahorro["van_eur"], 0),
        "tir_pct":         round(metricas_solo_ahorro["tir"] * 100, 2) if
metricas_solo_ahorro["tir"] else None,
        "payback_anios":   metricas_solo_ahorro["payback_anios"] if
np.isfinite(metricas_solo_ahorro["payback_anios"]) else None,
        "dpp_anios":       metricas_solo_ahorro["dpp_anios"] if
np.isfinite(metricas_solo_ahorro["dpp_anios"]) else None,
        "pi":              round(metricas_solo_ahorro["pi"], 3),
        "nota":            "Hipotesis conservadora: sin asumir ingresos
por compute IT",
    },
    {

```

```

        " analisis":          "M6 incremental sobre M4 (solo tanque +
integracion)",
        "capex_eur":          capex_incremental_m6,
        "beneficio_eur_ano":  round(beneficio_incr_m6, 0),
        "van_eur":            round(metricas_incr["van_eur"], 0),
        "tir_pct":            round(metricas_incr["tir"] * 100, 2) if
metricas_incr["tir"] else None,
        "payback_anios":      metricas_incr["payback_anios"] if
np.isfinite(metricas_incr["payback_anios"]) else None,
        "dpp_anios":          metricas_incr["dpp_anios"] if
np.isfinite(metricas_incr["dpp_anios"]) else None,
        "pi":                 round(metricas_incr["pi"], 3),
        "nota":               "Inversion marginal del tanque termico sobre M4
ya instalado",
    },
    {
        " analisis":          "M7 incremental sobre M6 (añadir bateria al
CTES)",
        "capex_eur":          capex_incr_m7m6,
        "beneficio_eur_ano":  round(beneficio_incr_m7m6, 0),
        "van_eur":            round(metricas_m7m6["van_eur"], 0),
        "tir_pct":            round(metricas_m7m6["tir"] * 100, 2) if
metricas_m7m6["tir"] else None,
        "payback_anios":      metricas_m7m6["payback_anios"] if
np.isfinite(metricas_m7m6["payback_anios"]) else None,
        "dpp_anios":          metricas_m7m6["dpp_anios"] if
np.isfinite(metricas_m7m6["dpp_anios"]) else None,
        "pi":                 round(metricas_m7m6["pi"], 3),
        "nota":               "¿Merece la pena añadir la bateria una vez
instalado el tanque?",
    },
    {
        " analisis":          "M7 incremental sobre M5 (añadir tanque a la
bateria)",
        "capex_eur":          capex_incr_m7m5,
        "beneficio_eur_ano":  round(beneficio_incr_m7m5, 0),
        "van_eur":            round(metricas_m7m5["van_eur"], 0),
        "tir_pct":            round(metricas_m7m5["tir"] * 100, 2) if
metricas_m7m5["tir"] else None,
        "payback_anios":      metricas_m7m5["payback_anios"] if
np.isfinite(metricas_m7m5["payback_anios"]) else None,

```

```
"dpp_anios":          metricas_m7m5["dpp_anios"] if
np.isfinite(metricas_m7m5["dpp_anios"]) else None,
    "pi":              round(metricas_m7m5["pi"], 3),
    "nota":            "¿Merece la pena añadir el tanque una vez
instalada la bateria?",
    },
])

# =====
# 12. IMPRIMIR RESUMEN
# =====

SEP = "=" * 84

print(f"\n{SEP}")
print("COMPARACION DE LOS 7 MODELOS")
print(SEP)
print(f"{'Modelo':<32} {'Coste anual':>13} {'Ahorro':>12} {'Ahorro %':>9}")
print("-" * 72)
for nombre, coste in costes.items():
    ah = coste_base - coste
    print(f"  {nombre:<30} {coste:>13,.0f}EUR {ah:>12,.0f}EUR
{ah/coste_base*100:>8.1f}%")

print(f"\n  M5 beneficio (solo arbitraje elec): {ahorro_energia_m5:,.0f}
EUR/ano")
print(f"  M6 beneficio total (energia + IT): {ben_m6:,.0f} EUR/ano")
print(f"  M7 beneficio total (energia + IT): {beneficio_total_m7:,.0f}
EUR/ano")

print(f"\n{SEP}")
print("ANALISIS DE INVERSION – Escenario base (0% crecimiento)")
print(SEP)
print(f"{'Modelo':<32} {'CAPEX':>12} {'VAN':>13} {'TIR':>7} {'PB':>5}
{'DPP':>5} {'PI':>6} {'OK':>4}")
print("-" * 84)
filt = df_modelos[df_modelos["escenario_precio"] == "Sin crecimiento (0%)"]
for _, r in filt.iterrows():
    tir_s = f"{r['tir_pct']:.1f}%" if pd.notna(r["tir_pct"]) else "  N/A"
    pb_s = f"{r['payback_simple_anios']:.0f}a" if
pd.notna(r["payback_simple_anios"]) else " inf"
```

```
dpp_s = f"{r['payback_descontado_anios']:.0f}a" if
pd.notna(r["payback_descontado_anios"]) else " inf"
pi_s = f"{r['indice_rentabilidad_pi']:.2f}" if
pd.notna(r["indice_rentabilidad_pi"]) else " N/A"
cap_s = f"{r['capex_eur']:>10,.0f}EUR" if r["capex_eur"] > 0 else
" sin CAPEX"
print(f" {r['modelo'][:30]:<30} {cap_s} {r['van_eur']:>13,.0f}EUR "
      f"{tir_s:>7} {pb_s:>5} {dpp_s:>5} {pi_s:>6} {r['viable']:>4}")

print(f"\n{SEP}")
print("LCOE POR COMPONENTE (EUR/MWh)")
print(SEP)
print(f"{'Modelo':<32} {'Capital':>8} {'O&M':>7} {'Energia':>8} {'Total':>8}
{'Ah.Energ':>9} {'Val.IT':>8}")
print("-" * 84)
for _, r in df_lcoe.iterrows():
    print(f" {r['modelo'][:30]:<30} {r['lcoe_capital_eur_mwh']:>8.2f} "
          f"{r['lcoe_om_eur_mwh']:>7.2f} {r['lcoe_energia_eur_mwh']:>8.2f} "
          f"{r['lcoe_total_eur_mwh']:>8.2f} "
          f"{r['ahorro_energia_eur_mwh']:>9.2f} "
          f"{r['valor_capacidad_it_eur_mwh']:>8.2f}")

print(f"\n{SEP}")
print("S1 – VAN M6 (EUR) | Coste tanque x Variacion beneficio")
print(SEP)
print(sens_s1.to_string())

print(f"\n{SEP}")
print("S2 – VAN M6 (EUR) | WACC x Crecimiento precio energia")
print(SEP)
print(sens_s2.to_string())

print(f"\n{SEP}")
print("S3 – VAN M6 (EUR) | Precio compute x Factor uso capacidad IT")
print(SEP)
print(sens_s3.to_string())

print(f"\n{SEP}")
print("S4 – VAN M6 (EUR) | Variacion beneficio x Variacion CAPEX")
print(SEP)
print(sens_s4.to_string())
```

```

print(f"\n{SEP}")
print("TORNADO CHART – Impacto de cada parametro en VAN M6")
print(SEP)
print(f"  VAN base M6: {VAN_BASE_M6:,.0f} EUR   |   VAN base M5:
{VAN_BASE_M5:,.0f} EUR")
print(f"  {'Parametro':<40} {'VAN low':>13} {'VAN high':>13} {'Rango':>13}")
print("-" * 84)
for _, r in df_tornado.iterrows():
    print(f"  {r['parametro']:<40} {r['van_low_eur']:>13,.0f}EUR "
          f"{r['van_high_eur']:>13,.0f}EUR
{r['rango_total_eur']:>13,.0f}EUR")

print(f"\n{SEP}")
print("BREAKEVEN – M5 (BESS) y M6 (CTES)")
print(SEP)
for _, r in df_breakeven.iterrows():
    print(f"  {r['analisis']}")
    print(f"    Umbral: {r['breakeven']} | Actual:
{r['actual']} | Margen: {r['margen']}")
    print()

print(f"\n{SEP}")
print("ANALISIS COMPLEMENTARIO – M6 (CTES)")
print(SEP)
ma = metricas_solo_ahorro
mi = metricas_incr
print(f"  A) M6 solo ahorro energetico (sin valor capacidad IT):")
print(f"    CAPEX: {capex_m6:,.0f} EUR | Beneficio:
{ahorro_energia_m6:,.0f} EUR/ano")
if ma["tir"]:
    print(f"    VAN: {ma['van_eur']:, .0f} EUR | TIR:
{ma['tir']*100:.1f}% | "
          f"    f\"PB: {ma['payback_anios']:.0f}a | DPP:
{ma['dpp_anios']:.0f}a | PI: {ma['pi']:.2f}\"")
print(f"    -> Proyecto VIABLE incluso sin asumir ingresos por compute IT")
print(f"\n  B) M6 incremental sobre M4 (tanque + integracion):")
print(f"    CAPEX incr.: {capex_incremental_m6:,.0f} EUR "
      f"    f\"(tanque {capex_tanque:,.0f} + integracion
{coste_integracion:,.0f})")
print(f"    Beneficio incr.: {beneficio_incr_m6:,.0f} EUR/ano "

```

```
f"(ahorro extra {ahorro_energia_m6 - ahorro_energia_m4:,.0f} + IT
{valor_capacidad_eur_ano:,.0f})")
if mi["tir"]:
    print(f"      VAN incr.: {mi['van_eur']:,.0f} EUR | TIR:
{mi['tir']*100:.1f}% | "
        f"PB: {mi['payback_anios']:,.0f}a | PI: {mi['pi']:0.2f}")
print(f"      -> El tanque justifica la inversion por si solo")

mc_ = metricas_m7m6
md_ = metricas_m7m5
print(f"\n C) M7 incremental sobre M6 (añadir bateria al CTES):")
print(f"      CAPEX incr.: {capex_incr_m7m6:,.0f} EUR | Beneficio incr.:
{beneficio_incr_m7m6:,.0f} EUR/ano")
tir_c = f"{mc_['tir']*100:.1f}%" if mc_["tir"] else "N/A"
print(f"      VAN incr.: {mc_['van_eur']:,.0f} EUR | TIR: {tir_c}")
print(f"      -> {'La bateria SI anade valor sobre el CTES' if mc_['van_eur']
> 0 else 'La bateria NO anade valor suficiente sobre el CTES'}")

print(f"\n D) M7 incremental sobre M5 (añadir tanque a la bateria):")
print(f"      CAPEX incr.: {capex_incr_m7m5:,.0f} EUR | Beneficio incr.:
{beneficio_incr_m7m5:,.0f} EUR/ano")
tir_d = f"{md_['tir']*100:.1f}%" if md_["tir"] else "N/A"
print(f"      VAN incr.: {md_['van_eur']:,.0f} EUR | TIR: {tir_d}")
print(f"      -> {'El tanque SI anade valor sobre la bateria' if
md_['van_eur'] > 0 else 'El tanque NO anade valor suficiente sobre la
bateria'}")

# =====
# 13. EXPORTAR EXCEL
# =====

try:
    _wr = pd.ExcelWriter(" analisis_economico_tfg.xlsx", engine="openpyxl")
except PermissionError:
    raise SystemExit(
        "\n[ERROR] No se puede escribir el Excel – cierra
' analisis_economico_tfg.xlsx' primero.\n"
    )

with _wr as writer:
```

```
df_modelos.to_excel(writer, sheet_name="comparacion_modelos",
index=False)

df_modelos[df_modelos["escenario_precio"] == "Sin crecimiento (0%)"] \
.to_excel(writer, sheet_name="resumen_base_0pct", index=False)

df_lcoe.to_excel(writer, sheet_name="lcoe_desglose", index=False)

# Flujos de caja por modelo (escenario 0%)
for nombre in costes:
    clave = f"{nombre} | Sin crecimiento (0%)"
    if clave in tablas_flujos:
        sname = nombre[:28].strip().replace("-", "").replace(" ", "_")
        tablas_flujos[clave].to_excel(writer, sheet_name=sname,
index=False)

# Flujos M6 todos los escenarios
for clave, df_fl in tablas_flujos.items():
    if "CTES" in clave:
        tablas_flujos[clave].to_excel(writer, sheet_name=clave[:31],
index=False)

sens_s1.to_excel(writer, sheet_name="S1_tanque_beneficio")
sens_s2.to_excel(writer, sheet_name="S2_wacc_precio")
sens_s3.to_excel(writer, sheet_name="S3_compute_uso")
sens_s4.to_excel(writer, sheet_name="S4_beneficio_capex")

df_tornado.to_excel(writer, sheet_name="tornado", index=False)

df_breakeven.to_excel(writer, sheet_name="breakeven_resumen",
index=False)
df_analisis_compl.to_excel(writer, sheet_name="analisis_complementario",
index=False)

df_curva_ben.to_excel(writer, sheet_name="curva_van_beneficio",
index=False)
df_curva_cap.to_excel(writer, sheet_name="curva_van_capex",
index=False)
df_curva_wacc.to_excel(writer, sheet_name="curva_van_wacc",
index=False)
```

```
df_curva_crec.to_excel(writer, sheet_name="curva_van_crec",
index=False)
df_curva_pcomp.to_excel(writer, sheet_name="curva_van_compute",
index=False)

hipotesis = pd.DataFrame([
    # Chiller
    {"parametro": "potencia_chiller_kw", "valor":
potencia_chiller_kw, "unidad": "kW", "fuente": "Modelo
M4/M5/M6"},
    {"parametro": "coste_chiller_eur_kw", "valor":
coste_chiller_eur_kw, "unidad": "EUR/kW", "fuente": "Mercado
industrial 2024"},
    # BESS (M5)
    {"parametro": "capacidad_bateria_kwh", "valor":
capacidad_bateria_kwh, "unidad": "kWh", "fuente": "Modelo BESS
M5"},
    {"parametro": "coste_bateria_eur_kwh", "valor":
coste_bateria_eur_kwh, "unidad": "EUR/kWh", "fuente": "Li-ion 2024
(BNEF)"},
    {"parametro": "capex_m5_total_eur", "valor":
capex_m5, "unidad": "EUR", "fuente": "Calculado"},
    {"parametro": "ahorro_energia_m5_eur_ano", "valor":
round(ahorro_energia_m5, 0), "unidad": "EUR/ano", "fuente": "Coste base -
Coste M5"},
    {"parametro": "van_base_m5_eur", "valor":
round(VAN_BASE_M5, 0), "unidad": "EUR", "fuente": "Calculado"},
    # CTES (M6)
    {"parametro": "capacidad_tanque_kwh_th", "valor":
capacidad_tanque_kwh_th, "unidad": "kWh_th", "fuente": "Modelo CTES
M6"},
    {"parametro": "coste_tanque_eur_kwh_th", "valor":
coste_tanque_eur_kwh_th, "unidad": "EUR/kWh_th", "fuente": "ASHRAE /
tanque industrial"},
    {"parametro": "capex_m6_total_eur", "valor":
capex_m6, "unidad": "EUR", "fuente": "Calculado"},
    {"parametro": "energia_it_base_kwh_ano", "valor":
round(energia_it_base_kwh, 0), "unidad": "kWh/ano", "fuente": "Suma col.
capacidad_it_liberada_kw"}],
```

```

        {"parametro": "potencia_it_media_kw", "valor":
round(potencia_it_media_kw,1),"unidad": "kW_e", "fuente": "Resultados
M6 (informativo)"},
        {"parametro": "horas_descarga_anual", "valor":
int(horas_descarga_anual), "unidad": "h/ano", "fuente": "Resultados
M6 (informativo)"},
        {"parametro": "precio_compute_eur_mwh", "valor":
precio_compute_eur_mwh, "unidad": "EUR/MWh", "fuente": "Contratos
colocation HPC/IA 2024"},
        {"parametro": "factor_uso_capacidad", "valor":
factor_uso_capacidad, "unidad": "pu", "fuente": "Hipotesis
conservadora"},
        {"parametro": "valor_capacidad_it_eur_ano", "valor":
round(valor_capacidad_eur_ano,0),"unidad": "EUR/ano", "fuente":
"energia_it_base x factor x precio"},
        {"parametro": "ahorro_energia_m6_eur_ano", "valor":
round(ahorro_energia_m6, 0), "unidad": "EUR/ano", "fuente": "Coste base -
Coste M6"},
        {"parametro": "beneficio_total_m6_eur_ano", "valor":
round(beneficio_total_m6, 0), "unidad": "EUR/ano", "fuente": "Calculado"},
        {"parametro": "van_base_m6_eur", "valor":
round(VAN_BASE_M6, 0), "unidad": "EUR", "fuente": "Calculado"},
        # Hibrido (M7)
        {"parametro": "capex_m7_total_eur", "valor":
capex_m7, "unidad": "EUR", "fuente": "Chiller +
bateria + tanque + integracion"},
        {"parametro": "ahorro_energia_m7_eur_ano", "valor":
round(ahorro_energia_m7, 0), "unidad": "EUR/ano", "fuente": "Coste base -
Coste M7"},
        {"parametro": "energia_it_m7_kwh_ano", "valor":
round(energia_it_m7_kwh, 0), "unidad": "kWh/ano", "fuente": "Suma col.
capacidad_it_liberada_kw (M7)"},
        {"parametro": "valor_capacidad_it_m7_eur_ano", "valor":
round(valor_capacidad_m7_ano, 0), "unidad": "EUR/ano", "fuente":
"energia_it_m7 x factor x precio"},
        {"parametro": "beneficio_total_m7_eur_ano", "valor":
round(beneficio_total_m7, 0), "unidad": "EUR/ano", "fuente": "Calculado"},
        # General
        {"parametro": "vida_util_anos", "valor":
vida_util, "unidad": "anos", "fuente": "Horizonte
TFG"},

```

```
        {"parametro": "tasa_descuento_wacc",          "valor":  
tasa_descuento,          "unidad": "decimal",      "fuente": "Hipotesis  
academica"},  
        {"parametro": "valor_residual_pct",          "valor":  
valor_residual_pct,      "unidad": "decimal",      "fuente": "Hipotesis  
conservadora"},  
        {"parametro": "capex_incremental_m6_eur",    "valor":  
capex_incremental_m6,    "unidad": "EUR",          "fuente": "Tanque +  
integracion"},  
    ])  
    hipotesis.to_excel(writer, sheet_name="hipotesis", index=False)  
  
print(f"\nArchivo generado: analisis_economico_tfg.xlsx")  
  
# =====  
# 14. EXPORTAR CSV  
# =====  
  
CSV_DIR = "csv_exports"  
os.makedirs(CSV_DIR, exist_ok=True)  
  
df_modelos.to_csv(f"{CSV_DIR}/comparacion_modelos.csv",      index=False,  
encoding="utf-8-sig")  
df_lcoe.to_csv(f"{CSV_DIR}/lcoe_desglose.csv",                index=False,  
encoding="utf-8-sig")  
df_tornado.to_csv(f"{CSV_DIR}/tornado.csv",                  index=False,  
encoding="utf-8-sig")  
df_breakeven.to_csv(f"{CSV_DIR}/breakeven_resumen.csv",       index=False,  
encoding="utf-8-sig")  
df_analisis_compl.to_csv(f"{CSV_DIR}/analisis_complementario.csv",  
index=False, encoding="utf-8-sig")  
df_curva_ben.to_csv(f"{CSV_DIR}/curva_van_beneficio.csv",     index=False,  
encoding="utf-8-sig")  
df_curva_cap.to_csv(f"{CSV_DIR}/curva_van_capex.csv",         index=False,  
encoding="utf-8-sig")  
df_curva_wacc.to_csv(f"{CSV_DIR}/curva_van_wacc.csv",         index=False,  
encoding="utf-8-sig")  
df_curva_crec.to_csv(f"{CSV_DIR}/curva_van_crec.csv",         index=False,  
encoding="utf-8-sig")  
df_curva_pcomp.to_csv(f"{CSV_DIR}/curva_van_compute.csv",    index=False,  
encoding="utf-8-sig")
```

```
sens_s1.to_csv(f"{CSV_DIR}/S1_tanque_beneficio.csv", encoding="utf-8-sig")
sens_s2.to_csv(f"{CSV_DIR}/S2_wacc_precio.csv", encoding="utf-8-sig")
sens_s3.to_csv(f"{CSV_DIR}/S3_compute_uso.csv", encoding="utf-8-sig")
sens_s4.to_csv(f"{CSV_DIR}/S4_beneficio_capex.csv", encoding="utf-8-sig")
hipotesis.to_csv(f"{CSV_DIR}/hipotesis.csv", index=False, encoding="utf-8-sig")

df_modelos[df_modelos["escenario_precio"] == "Sin crecimiento (0%)"] \
    .to_csv(f"{CSV_DIR}/resumen_base_0pct.csv", index=False, encoding="utf-8-sig")

csv_files = sorted(os.listdir(CSV_DIR))
print(f"CSVs generados en '{CSV_DIR}/' ({len(csv_files)} archivos)")
```

8.9 ARCHIVOS DE DATOS Y RESULTADOS COMPLEMENTARIOS

Por su volumen (perfiles horarios de 8.760 registros por modelo), los archivos de datos y resultados completos no se incluyen impresos en este anexo. Se alojan en una carpeta compartida de SharePoint de la Universidad Pontificia Comillas, accesible desde el siguiente enlace:

[Carpeta compartida — Archivos de datos y resultados \(TFG\)](#)

https://upcomillas-my.sharepoint.com/:f/g/personal/202110425_alu_comillas_edu/IgDmghhmwP3KSYCSDrIYySe5ATbJVv-tjLBRfKVTRyaGdaU?e=61nwcC

Contenido de la carpeta

Archivo	Contenido
analisis_economico_tfg.xlsx	Análisis económico completo: comparación de modelos, VAN/TIR, sensibilidad, hipótesis
Apilado_de_datos.xlsx	Datos horarios de todos los modelos apilados en una única tabla (formato largo)

Archivo	Contenido
Completo.xlsx	Libro completo con los resultados de todos los modelos integrados
datos_modelo.xlsx	Perfil de demanda horaria del CPD generado para el caso base (8.760 h)
descargar_temperatura_openmeteo.py	Script en Python usado para descargar los datos de temperatura de Open-Meteo
graficas resumen.xlsx	Gráficas resumen utilizadas en la memoria y la presentación
resultados_modelo_1_base.xlsx	Resultados horarios del Modelo 1 — Caso base
resultados_modelo_2_training.xlsx	Resultados horarios del Modelo 2 — Flexibilidad de training anual
resultados_modelo_3_training_diario.xlsx	Resultados horarios del Modelo 3 — Flexibilidad de training diaria
resultados_modelo_4_clima.xlsx	Resultados horarios del Modelo 4 — Modelo climático y refrigeración
resultados_modelo_5_final.xlsx	Resultados horarios del Modelo 5 — Sistema con batería (BESS)
resultados_modelo_6_ctes.xlsx	Resultados horarios del Modelo 6 — Sistema CTES
resultados_modelo_7_hibrido.xlsx	Resultados horarios del Modelo 7 — Sistema híbrido BESS+CTES
temperatura_alcobendas_2025.xlsx	Datos horarios de temperatura exterior (Open-Meteo, Alcobendas)

El acceso está restringido a cuentas de la Universidad Pontificia Comillas (@comillas.edu), sin fecha de expiración.