

TABLA DE CONTENIDOS

2	INTRODUCCIÓN	2
3	DESCRIPCIÓN DEL PROBLEMA FÍSICO	2
4	GENERACIÓN DE DATASET	3
4.1	Diseño de las simulaciones	3
4.2	Estructura del dataset	3
5	ENFOQUE DE MACHINE LEARNING	4
5.1	Arquitectura y funcionamiento de la red neuronal para predicción de temperatura	4
5.2	Configuración de entrenamiento	4
6	RESULTADOS	5
6.1	Modelo con dataset amplio – Sin ponderación	5
6.2	Modelo con dataset amplio – Con ponderación	8
7	CONCLUSIONES	11
8	ANEXO	13
8.1	Ecuaciones – Método de las Diferencias Finitas	13
8.2	Registro de prueba, hiperparámetros y archivos asociados	15

2 INTRODUCCIÓN

En los últimos años, las técnicas de Machine Learning (ML) se han consolidado como herramientas eficaces para resolver problemas complejos en ciencia e ingeniería, particularmente aquellos que involucran sistemas físicos descritos por ecuaciones diferenciales. Su capacidad para aprender patrones no lineales a partir de grandes volúmenes de datos simulados o experimentales ha permitido abordar problemas tradicionalmente tratados mediante métodos numéricos, con un enfoque más flexible, veloz y escalable.

Uno de los ámbitos donde esta sinergia resulta especialmente prometedora es el estudio de fenómenos de conducción térmica. En estos sistemas, las condiciones de contorno, la geometría y la presencia de fuentes de calor localizadas, pueden producir distribuciones de temperatura altamente complejas. La posibilidad de entrenar modelos de ML sobre soluciones numéricas precisas, obtenidas mediante métodos como diferencias finitas, abre la puerta al desarrollo de predictores térmicos eficientes, capaces de reemplazar (o complementar) los métodos tradicionales con una reducción significativa en tiempos de cómputo.

En este contexto, el presente trabajo explora el desarrollo de un modelo de ML para predecir la distribución de temperaturas en una **chapa cuadrada**, considerando distintas combinaciones de **condiciones de borde** (temperatura o flujo) y la presencia de un **punto caliente localizado**. Para ello, se generó un conjunto de datos mediante simulaciones por diferencias finitas, el cual se utilizó para entrenar una red neuronal multicapa (MLP).

El objetivo de esta etapa es evaluar la viabilidad del enfoque propuesto, a través de la implementación y entrenamiento de un modelo preliminar, el análisis de su desempeño en términos de error de predicción y capacidad de generalización, y la identificación de oportunidades de mejora tanto en la arquitectura de la red como en el diseño del dataset. Esta primera aproximación busca sentar las bases para avanzar hacia un modelo robusto y confiable, capaz de abordar con eficacia el problema térmico planteado.

3 DESCRIPCIÓN DEL PROBLEMA FÍSICO

El problema físico consiste en determinar la distribución de temperaturas en una chapa cuadrada de dimensiones conocidas, considerando distintas condiciones de borde y la presencia de un punto caliente. El comportamiento térmico se modela utilizando la ecuación de conducción de calor en estado estacionario en dos dimensiones:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

La resolución se realiza mediante el método de diferencias finitas, discretizando la chapa en una malla de $N_x \times N_y$ nodos con pasos Δx y Δy determinados según las dimensiones y la resolución buscada. La resolución del sistema resulta en un conjunto de ecuaciones lineales de la forma:

$$A T = b$$

donde T es el vector que contiene las temperaturas en cada nodo de la malla.

Las condiciones de contorno consideradas corresponden a los siguientes tipos:

- Condición de temperatura (Dirichlet): se fija el valor de temperatura en el borde.
- Condición de flujo (Neumann): se fija el flujo de calor a través del borde.
- Combinaciones entre ambas, variando según el caso simulado.

Además, se incluye un punto caliente que actúa como una fuente puntual de calor, con coordenadas discretas (i, j) dentro de la malla y un valor de temperatura definido.

Los parámetros físicos considerados en el modelo son:

- Conductividad térmica (k), constante para cada simulación.
- Tipo y valor de condiciones de contorno en cada borde.
- Posición y temperatura del punto caliente.

El detalle completo del esquema de resolución numérica se encuentra desarrollado en el **ANEXO 7.1**.

4 GENERACIÓN DE DATASET

Para entrenar el modelo de predicción, se construyó un conjunto de datos sintético basado en simulaciones numéricas del problema térmico descripto. Este conjunto representa distintos escenarios físicos generados mediante la variación sistemática de parámetros y condiciones.

4.1 Diseño de las simulaciones

Cada simulación se configuró mediante la selección aleatoria de:

- Condiciones de contorno en cada borde (A, B, C, D), seleccionando aleatoriamente si la condición es de temperatura o flujo, y asignando un valor dentro de un rango físico coherente:
 - Temperatura: desde -196°C hasta $T_{\text{fusion}} - 10^{\circ}\text{C}$ del material seleccionado.
 - Flujo: entre $-10.000 \frac{\text{W}}{\text{m}^2}$ y $10.000 \frac{\text{W}}{\text{m}^2}$.
- Punto caliente, asignando una posición aleatoria dentro de la malla y una temperatura aleatoria dentro del rango del material.
- Material base, elegido desde una base de datos que incluye propiedades térmicas, en particular la conductividad térmica y la temperatura de fusión.

Cada combinación generada define un caso térmico único, cuya solución se obtiene resolviendo el sistema estacionario por diferencias finitas, con los parámetros físicos y condiciones de borde correspondientes.

4.2 Estructura del dataset

Una vez resuelto el sistema para cada caso, se construyen los siguientes elementos:

- Un vector de entrada que contiene las variables relevantes del sistema: tipo y valor de condiciones de borde, material seleccionado, propiedades térmicas, y datos del punto caliente (**12 variables móviles**)
- Un vector de salida, correspondiente a la distribución de temperaturas en toda la malla.
- Un registro asociado, que permite identificar cada muestra y analizar sus características físicas.

Para garantizar la validez física del dataset, se aplicó un filtro posterior que elimina aquellas simulaciones que presentan temperaturas fuera del rango admisible.

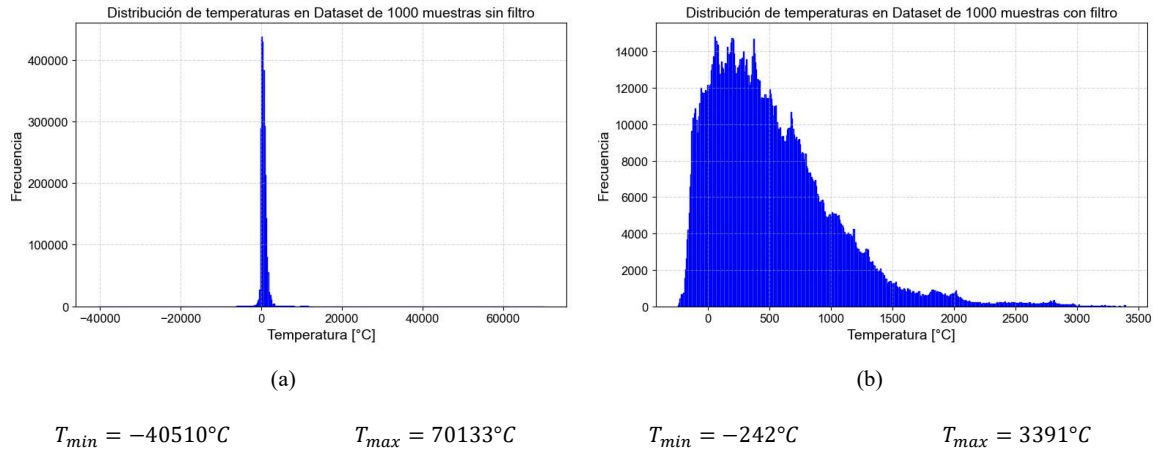


Figura 1: (a) Histograma de temperaturas para 1000 muestras sin aplicar filtro, considerando condiciones de flujo y temperatura en el rango de $[-10.000 \frac{\text{W}}{\text{m}^2}, 10.000 \frac{\text{W}}{\text{m}^2}]$ y $[-216^{\circ}\text{C}, T_{\text{Fusion}} - 10^{\circ}\text{C}]$. (b) Histograma de temperaturas para las mismas 1000 muestras tras aplicar un filtro que descarta aquellas muestras que presentan temperaturas $T_{i,j} < -250^{\circ}\text{C}$ o $T_{i,j} > (T_{\text{fusion}} - 10)$.

5 ENFOQUE DE MACHINE LEARNING

5.1 Arquitectura y funcionamiento de la red neuronal para predicción de temperatura

Según lo descrito anteriormente, el sistema dispone de 12 variables móviles que pueden variar dentro de diferentes rangos. La salida está compuesta por 2500 nodos, cada uno representando la temperatura de un punto específico de la placa.

Para modelar esta relación, se implementó una arquitectura que recibe las 12 variables de entrada y devuelve las 2500 temperaturas correspondientes. En este caso, se utiliza una red neuronal multicapa (MLP) con tres capas ocultas, pensada para capturar la relación no lineal entre las variables de entrada y la distribución de temperaturas en la placa.

La arquitectura está conformada por:

- Una capa de entrada de 12 nodos, correspondientes a las variables móviles del sistema.
- Una primera capa oculta con 128 neuronas y función de activación ReLU.
- Una segunda capa oculta con 256 neuronas y función de activación ReLU.
- Una tercera capa oculta con 512 neuronas y función de activación ReLU.
- Una capa de salida con 2500 neuronas, correspondiente a cada punto de temperatura de la placa.

Esta estructura se implementa de forma secuencial, empleando funciones de activación ReLU en las capas ocultas buscando modelar de manera eficiente las no linealidades del sistema, mientras que la capa de salida se mantiene lineal para predecir valores de temperatura en una escala continua.

Cada neurona de la red neuronal realiza, de manera individual, el siguiente procedimiento: recibe señales de entrada correspondientes a las variables móviles o a las salidas de la capa anterior, multiplica cada entrada por un peso sináptico (w_1, w_2, \dots, w_n) que se ajusta durante el entrenamiento y suma un término independiente denominado bias (θ), que permite ajustar el nivel de activación de la neurona. Sobre este valor resultante se aplica la función de activación ReLU ($g(x)$) en las capas ocultas, lo que permite a la red capturar relaciones complejas entre las variables de entrada y la salida. Finalmente, la salida de cada neurona se transmite a la siguiente capa repitiendo nuevamente el proceso.

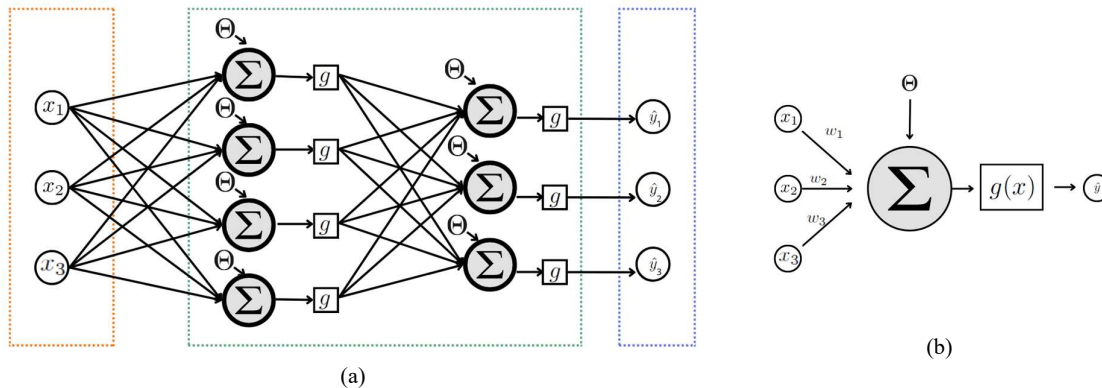


Figura 2: (a) Esquema de la arquitectura de la red neuronal multicapa (MLP) utilizada para predecir las temperaturas en cada uno de los 2500 puntos de la placa a partir de las 12 variables móviles del sistema. (b) Esquema del funcionamiento de una neurona dentro de la red, mostrando la combinación ponderada de las entradas, la adición del término bias y la aplicación de la función de activación

5.2 Configuración de entrenamiento

Para entrenar el modelo descrito en la sección anterior, se utilizaron los “datasets” generados mediante simulaciones por diferencias finitas, divididos en 70% para entrenamiento y 30% para validación, permitiendo evaluar la capacidad de generalización del modelo mientras ajusta sus parámetros.

Con el objetivo de facilitar el aprendizaje y asegurar estabilidad numérica, tanto las variables de entrada como las de salida fueron normalizadas respecto de su media (μ) y desvío estándar (σ), de acuerdo con las siguientes expresiones:

$$x^* = \frac{(x - \mu_x)}{\sigma_x}$$
$$T^* = \frac{T - \mu_T}{\sigma_T}$$

De este modo, las entradas y las salidas quedaron centradas en cero y escaladas, favoreciendo la eficiencia en el proceso de entrenamiento.

El entrenamiento de la red neuronal multicapa (MLP) se realizó durante un número definido de épocas, donde en cada época el conjunto de entrenamiento se recorre en lotes de muestras de tamaño fijo (batch size). Este enfoque permite actualizar los parámetros de la red de manera progresiva y estable, evitando los problemas que pueden surgir al utilizar todo el dataset en una única actualización.

En cada iteración sobre un lote de datos:

- La red recibe un conjunto de muestras normalizadas como entrada y calcula las temperaturas de salida estimadas
- Se calcula el error de predicción utilizando como métrica el Error Cuadrático Medio, definido como:

$$MSE = \frac{1}{N} \sum_{i=1}^N (\hat{T}_i^* - T_i^*)^2$$

Donde N representa la cantidad de puntos de salida (2500 en este caso) por muestra.

- A partir del cálculo de error, se computan las derivadas necesarias para conocer cómo deben ajustarse los pesos y bias de la red, y se actualizan los parámetros de forma tal de reducir el error en las siguientes iteraciones.

Una vez finalizada cada época, se evalúa el desempeño del modelo sobre el conjunto de validación, registrando nuevamente el MSE para monitorear el progreso del entrenamiento y prevenir el sobreajuste.

6 RESULTADOS

6.1 Modelo con dataset amplio – Sin ponderación

Con el objetivo de evaluar el desempeño del modelo a lo largo del entrenamiento, se monitoreó la evolución del **RMSE** (raíz del error cuadrático medio) tanto para el conjunto de entrenamiento como para el de validación. Esta métrica, al expresarse en las mismas unidades físicas que la salida ($^{\circ}\text{C}$), permite interpretar de forma directa la magnitud del error cometido por el modelo.

$$RMSE = \sqrt{MSE}$$

La **Figura 3** ilustra cómo varía el RMSE en función de la tasa de aprendizaje y la cantidad de épocas. Los resultados muestran que una tasa más baja favorece la reducción sostenida del error, en especial sobre el conjunto de validación, mientras que tasas más altas tienden a estabilizarse rápidamente en valores más elevados. Asimismo, prolongar el entrenamiento más allá de las 1000 épocas permite seguir disminuyendo el error, aunque con mejoras gradualmente menores, especialmente cuando se dispone de un conjunto de datos amplio.

En la **Figura 4**, se analiza el impacto del tamaño del dataset sobre la calidad del modelo. Se observa una mejora sustancial en el error tanto en entrenamiento como en validación al aumentar la cantidad de muestras, junto con una mayor estabilidad en las curvas. El modelo entrenado con 100.000 muestras logra una reducción significativa del RMSE final y evidencia un mejor comportamiento general frente a variaciones en las condiciones de borde y el punto caliente.

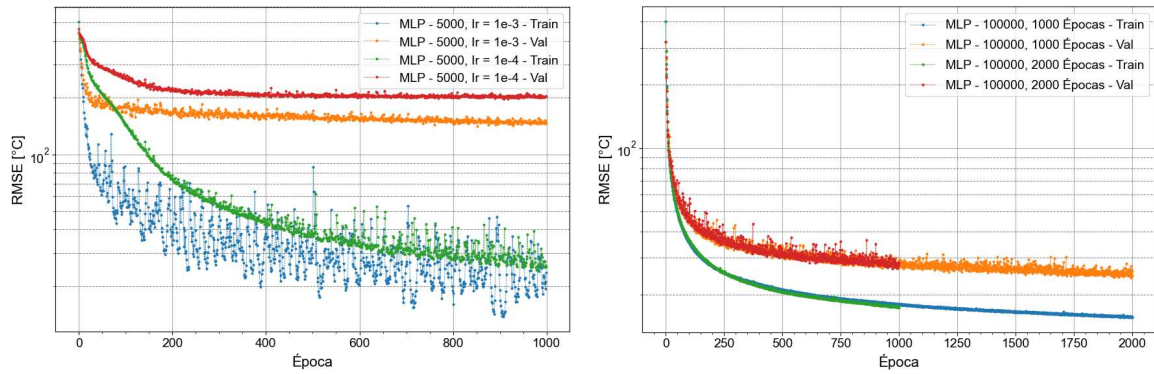


Figura 3: Evolución del RMSE durante el entrenamiento y la validación del modelo. **(a)** Comparación entre dos valores de tasa de aprendizaje ($lr = 10^{-3}$ y 10^{-4}) para un conjunto reducido de 5000 muestras. **(b)** Comparación entre dos configuraciones de cantidad de épocas (1000 y 2000) sobre un conjunto más amplio de 100.000 muestras

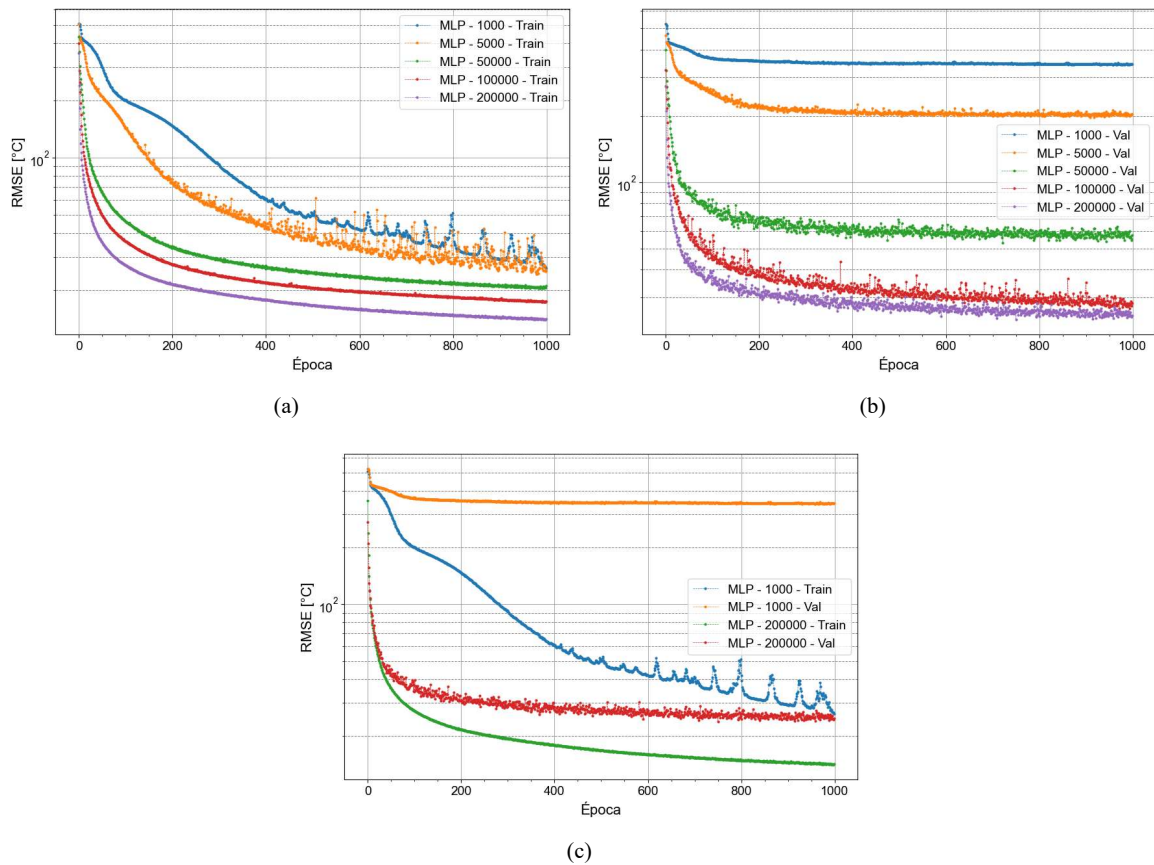


Figura 4: **(a)** Evolución del RMSE en el conjunto de entrenamiento para distintos tamaños de dataset, mostrando la reducción progresiva del error a lo largo de las épocas. **(b)** Evolución del RMSE en el conjunto de validación para los mismos tamaños de dataset, permitiendo evaluar la capacidad de generalización del modelo durante el entrenamiento. **(c)** Comparación directa entre los casos de menor y mayor tamaño de dataset, mostrando el efecto de la cantidad de muestras sobre la estabilidad y el valor final del error alcanzado.

En conjunto, los resultados de ambas figuras confirman que la calidad del modelo depende fuertemente de una elección adecuada de los hiperparámetros y de la cantidad de información disponible en el proceso de entrenamiento.

La **Figura 5** muestra un caso representativo en el que se comparan las distribuciones de temperatura obtenidas mediante diferencias finitas (referencia) y mediante el modelo de red neuronal entrenado. Se trata de una configuración sin fuente puntual, con gradientes térmicos moderados y condiciones mixtas en los bordes.

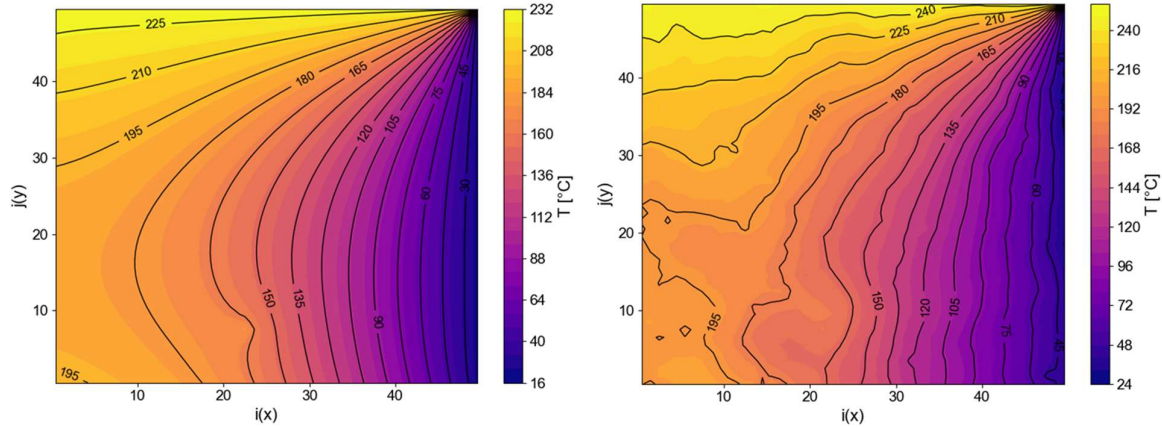


Figura 5: Comparación de la distribución de temperaturas obtenida con el modelo de diferencias finitas (izquierda) y el modelo de red neuronal MLP sin ponderación puntual del error (derecha). El modelo MLP fue entrenado con 200.000 muestras, tasa de aprendizaje $lr = 10^{-4}$ y 1000 épocas.

La predicción del modelo MLP replica con precisión el perfil térmico general y mantiene la forma de las isotermas en casi todo el dominio. La diferencia máxima registrada es de $24,2^{\circ}\text{C}$, mientras que la diferencia media y el desvío estándar se mantienen bajos ($12,3^{\circ}\text{C}$ y $4,76^{\circ}\text{C}$ respectivamente), lo que indica una buena consistencia en las predicciones. Estos resultados confirman que, en configuraciones suavemente variables, el modelo es capaz de aproximar correctamente la solución física, tanto de forma cualitativa como cuantitativa.

En contraste, la **Figura 6** presenta un caso más exigente, en el que se introduce un **punto caliente** como fuente localizada de temperatura. Esta condición genera un gradiente abrupto en una región acotada de la chapa, lo que representa un desafío específico para la red neuronal.

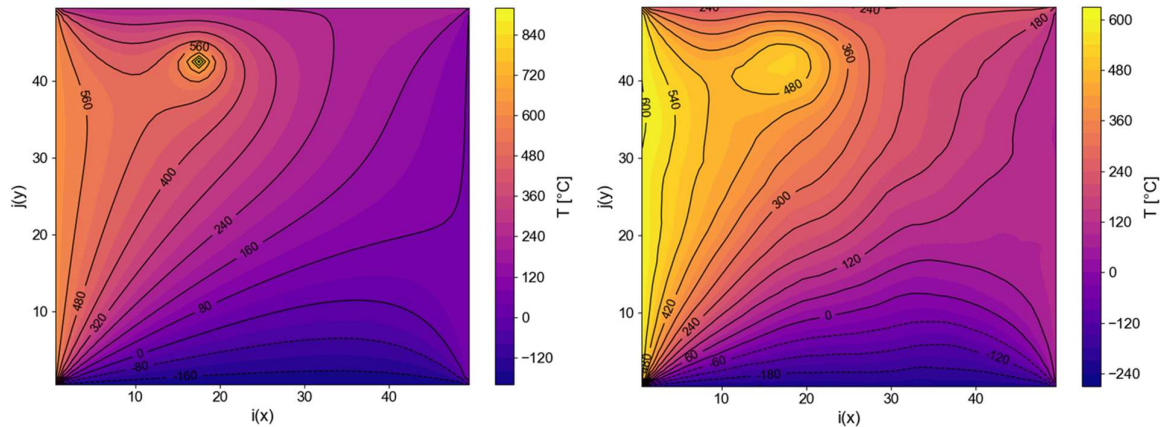


Figura 6: Comparación de la distribución de temperaturas para un caso con punto caliente. A la izquierda se muestra la solución de referencia obtenida mediante diferencias finitas, mientras que a la derecha se presenta la predicción del modelo MLP con ponderación, entrenado con 200.000 muestras, tasa de aprendizaje $lr = 10^{-4}$ y 1000 épocas.

Si bien el modelo logra capturar el comportamiento global del campo térmico, incluyendo las zonas de alta temperatura, su precisión disminuye notablemente en el entorno del punto caliente. Se observa una subestimación importante en la temperatura máxima alcanzada, lo que se refleja en un error absoluto

máximo de 400 °C. A nivel global, el modelo mantiene una diferencia media de 16,5 °C y un desvío estándar de 15,0 °C, valores claramente superiores a los obtenidos en el caso anterior.

Este resultado revela una limitación actual del enfoque, vinculada a la dificultad de generalizar correctamente en regiones con variaciones bruscas, como el entorno del punto caliente. Al tratarse de una cuestión de sensibilidad espacial, no basta con aumentar el volumen de datos o prolongar el entrenamiento. Por ello, se trabajó en una modificación de la función de pérdida, incorporando un esquema de ponderación espacial que permita focalizar el aprendizaje en el entorno del punto caliente.

6.2 Modelo con dataset amplio – Con ponderación

Con el objetivo de mejorar la precisión del modelo en el entorno del punto caliente, se implementó una estrategia de ponderación espacial del error durante el entrenamiento. Esta consiste en modificar la función de pérdida, introduciendo un término que pone mayor énfasis en las regiones cercanas al punto de alta temperatura.

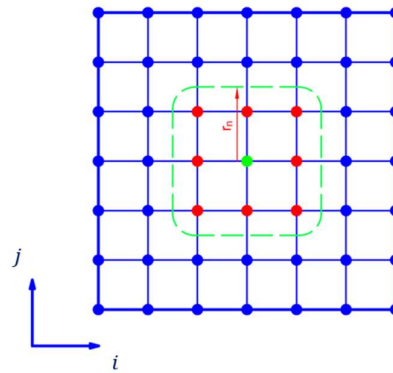


Figura 7: Esquema de la región de ponderación alrededor del punto caliente. El nodo central (en verde) representa la posición del punto caliente, mientras que los nodos dentro del radio r_n (en rojo) definen la zona crítica donde se aplica mayor peso en la función de pérdida

Tal como se ilustra en la **Figura 7**, el dominio de la chapa se divide en dos regiones:

- Una zona crítica centrada en el punto caliente (en verde), que abarca un radio definido por un parámetro r_n , e incluye un total de $N_{radio} = (2r_n + 1)^2$ nodos (en rojo).
- El resto del dominio, compuesto por los demás nodos de la malla.

Sobre cada una de estas regiones se calcula un error cuadrático medio independiente:

$$MSE_{radio} = \frac{1}{N_{radio}} \sum_{i=1}^{N_{radio}} (\hat{T}_{i,radio}^* - T_{i,radio}^*)^2$$

$$MSE_{resto} = \frac{1}{N_{resto}} \sum_{i=1}^{N_{resto}} (\hat{T}_{i,resto}^* - T_{i,resto}^*)^2$$

La función de pérdida total se define como una combinación ponderada de ambos términos:

$$MSE_{total} = (1 - \alpha) * MSE_{resto} + \alpha * MSE_{radio}$$

donde el parámetro $\alpha \in [0,1]$ permite ajustar el grado de prioridad a la región crítica. Esta formulación permite orientar explícitamente el aprendizaje del modelo hacia las zonas donde los errores son más relevantes desde el punto de vista físico, sin descuidar el comportamiento general del sistema.

La **Figura 8** muestra el comportamiento del error RMSE durante el entrenamiento de modelos con diferentes cantidades de muestras, aplicando la función de pérdida ponderada descrita anteriormente, con $\alpha = 0,3$ y radio $r_n = 2$. Se analizaron cinco tamaños de dataset (de 1000 a 200.000 muestras), manteniendo fijos el resto de los hiperparámetros.

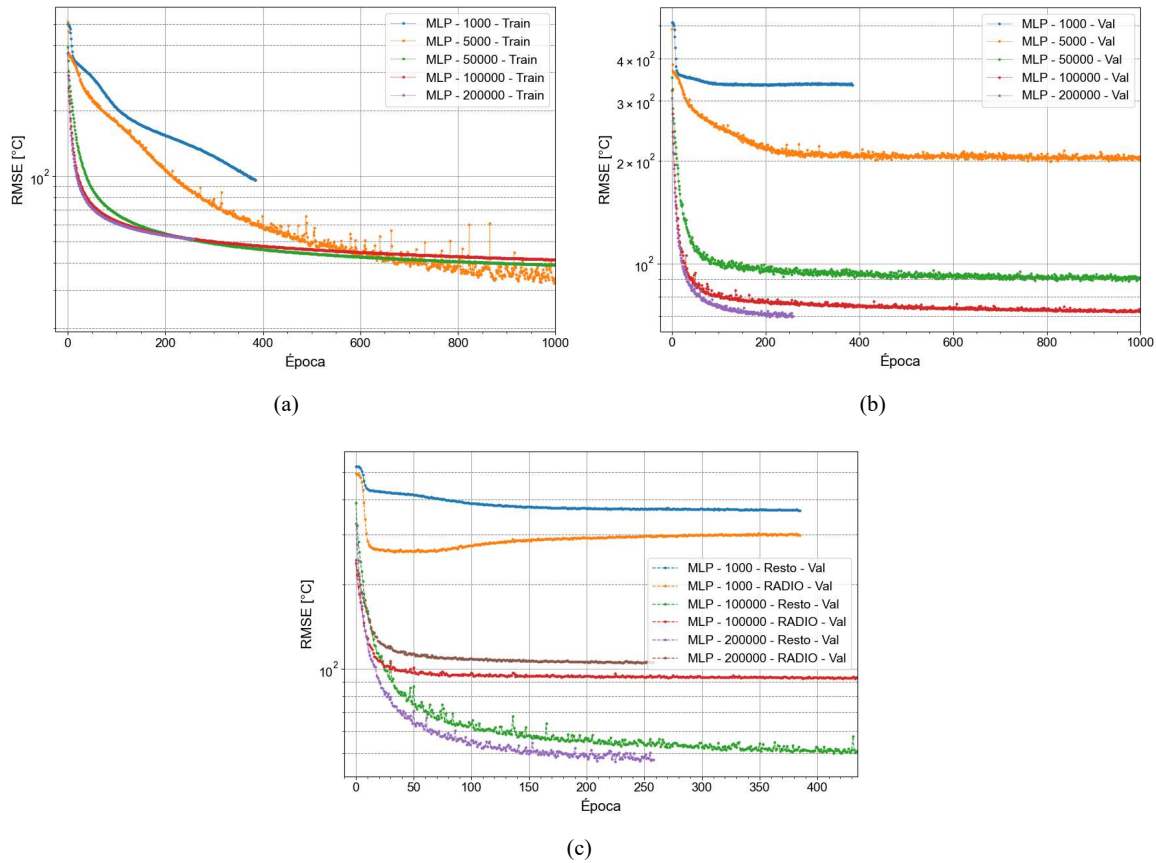


Figura 8: Evolución del error RMSE durante el entrenamiento de modelos MLP con función de pérdida ponderada, utilizando $\alpha = 0,3$ y radio $r_n = 2$. Se evalúa el impacto del tamaño del dataset sobre: **(a)** el conjunto de entrenamiento, **(b)** el conjunto de validación global, **(c)** la región ponderada (radio) y el resto de los nodos por separado.

En los gráficos **(a)** y **(b)** se observa que, a medida que se incrementa la cantidad de datos, el modelo no solo converge más rápido, sino que además logra un error notablemente menor tanto en entrenamiento como en validación global. La disminución del RMSE es significativa a partir de las 50.000 muestras, estabilizándose progresivamente para datasets más grandes. Esto confirma que la calidad de generalización mejora sustancialmente cuando se cuenta con un volumen adecuado de información, aún con la ponderación activa.

El gráfico **(c)** aporta un análisis más específico, desglosando el RMSE en la zona ponderada (radio) y en el resto del dominio. Se observa que los modelos entrenados con mayores volúmenes de datos logran reducir de forma considerable el error en la región crítica, sin deteriorar el desempeño en el resto. En particular, el modelo con 200.000 muestras muestra una mejora clara y sostenida en ambos sectores, lo cual valida la efectividad del esquema de ponderación para reforzar el aprendizaje localizado sin sacrificar precisión global.

Estos resultados muestran que la combinación de un dataset suficientemente amplio y una función de pérdida orientada espacialmente permite superar, al menos parcialmente, las limitaciones detectadas en el caso con punto caliente.

La **Figura 9** resume el comportamiento del error RMSE durante el entrenamiento de modelos MLP utilizando una función de pérdida ponderada con radio fijo $r_n = 1$, evaluando el impacto de distintas configuraciones de α y tamaños de dataset.

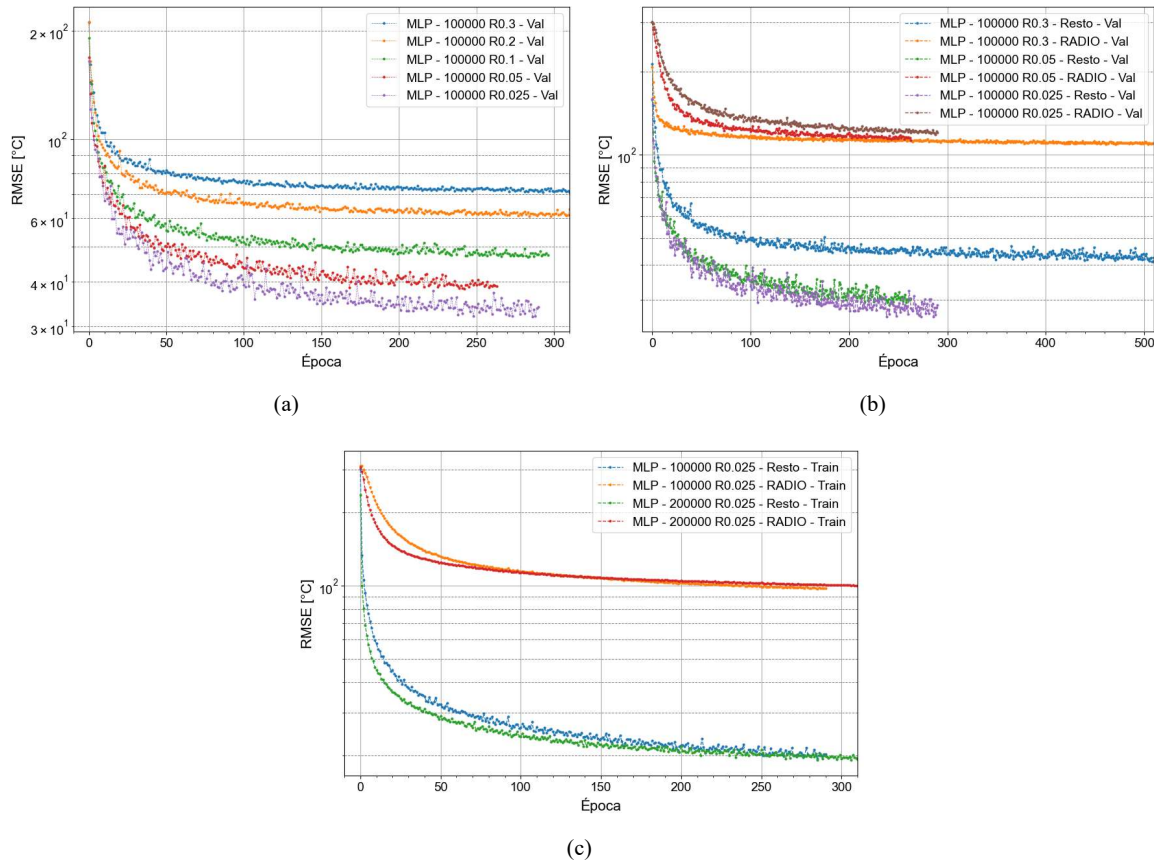


Figura 9: Evolución del error RMSE durante el entrenamiento y validación de modelos MLP con función de pérdida ponderada, utilizando un radio fijo de ponderación $r_n = 1$ **(a)** RMSE de validación global para un dataset de 100.000 muestras, considerando diferentes valores del parámetro de ponderación α . **(b)** RMSE de validación diferenciado por región (zona ponderada “radio” y resto del dominio), para distintos valores de α , manteniendo fijo el tamaño del dataset en 100.000 muestras. **(c)** Curvas de entrenamiento para los modelos con $\alpha = 0,025$, que muestran la estabilización del error al aumentar el tamaño del dataset, indicando que se ha alcanzado un límite en cuanto a mejora por cantidad de datos.

En el gráfico **(a)** se observa cómo disminuye el error de validación global al reducir el valor de α . Este comportamiento indica que, al priorizar el error general del dominio, el modelo logra una mejor capacidad de generalización. En cambio, valores elevados de α (que asignan mayor peso a la región crítica) tienden a degradar el desempeño global.

El gráfico **(b)** descompone el RMSE en dos regiones: el entorno del punto caliente (zona ponderada) y el resto del dominio. Se observa que, al aumentar α , el error en el resto crece, mientras que el error en el radio mejora inicialmente pero luego se estabiliza. Este último comportamiento, repetido en múltiples configuraciones, sugiere que el modelo alcanza un límite de mejora en la región crítica, más allá del cual no es posible seguir reduciendo el error, independientemente del valor de α .

El gráfico **(c)** explora el impacto del tamaño del dataset para un valor intermedio de ponderación (α). Se confirma que el error disminuye con el aumento de muestras, pero se estabiliza más allá de las 100.000, indicando que el modelo alcanza una meseta en su capacidad de aprendizaje con la configuración actual. La combinación de estos resultados sugiere que, si bien la ponderación espacial contribuye a mejorar la precisión en regiones críticas, existen límites estructurales que impiden resolver completamente el problema.

La **Figura 10** muestra que a medida que se incrementa la cantidad de muestras utilizadas para el entrenamiento, el modelo muestra mejoras notables en la reconstrucción de la distribución de temperaturas, reduciendo tanto el error medio como el máximo. Sin embargo, incluso con 200000 muestras, se mantienen

diferencias significativas en regiones puntuales, lo que indica que el enfoque mejora la sensibilidad local, pero no es suficiente por sí solo para capturar completamente las variaciones más abruptas del sistema.

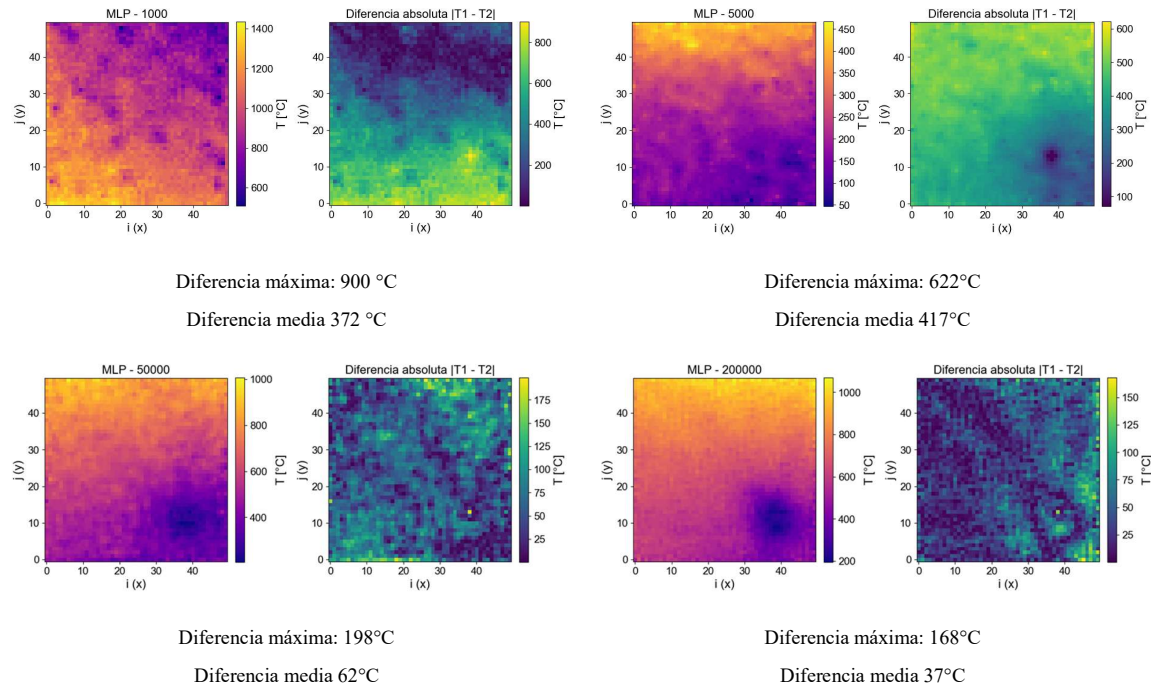


Figura 10: Comparación visual entre las predicciones del modelo MLP con ponderación y la solución de referencia, para diferentes tamaños del conjunto de entrenamiento (1000, 5000, 50000 y 200000 muestras). En cada caso se muestra la predicción generada por el modelo (izquierda) y el mapa de error absoluto con respecto a la solución por diferencias finitas (derecha). Se incluyen los valores de diferencia máxima y media obtenidos en cada muestra. Se observa una mejora progresiva con el aumento de datos, aunque persisten errores significativos en zonas críticas incluso con grandes volúmenes de entrenamiento

En conjunto, los resultados mostrados en esta sección permiten valorar el efecto concreto de la estrategia de ponderación sobre el entrenamiento. Se observa una mejora efectiva en la región crítica cuando se utiliza una ponderación moderada y un volumen de datos suficiente. Al mismo tiempo, el error en el resto del dominio también puede reducirse si se prioriza el desempeño global.

No obstante, la mejora tiene un límite. Aun incrementando la cantidad de muestras o ajustando los parámetros de ponderación, el error en la zona crítica tiende a estabilizarse. Esto indica que la estrategia, si bien útil, no es suficiente por sí sola para abordar por completo la sensibilidad local del problema.

Estas observaciones motivan la exploración de enfoques más avanzados que complementen la ponderación espacial, como arquitecturas más expresivas o esquemas de entrenamiento adaptativos.

7 CONCLUSIONES

En este trabajo se exploró el uso de técnicas de Machine Learning (ML) para predecir la distribución de temperaturas en una chapa cuadrada sometida a condiciones variadas de contorno y la presencia de un punto caliente. El enfoque adoptado, basado en una red neuronal tipo MLP entrenada con datos generados mediante simulaciones por diferencias finitas, logró una buena aproximación general al problema físico planteado, especialmente en escenarios con variaciones suaves de temperatura.

La implementación de una función de pérdida ponderada permitió orientar el aprendizaje del modelo hacia regiones críticas, mejorando la precisión local cerca del punto caliente. Sin embargo, los resultados también evidenciaron limitaciones claras: a pesar del ajuste en la ponderación espacial, el modelo alcanzó un límite en su capacidad para reducir errores en zonas de alto gradiente térmico, aun incrementando considerablemente el volumen del dataset de entrenamiento.

En conjunto, estos resultados destacan el potencial del uso combinado de simulaciones numéricas y ML para problemas de conducción térmica, pero también indican la necesidad de explorar estrategias adicionales. Arquitecturas más expresivas, enfoques híbridos y esquemas adaptativos de entrenamiento se presentan como alternativas para superar las limitaciones encontradas y mejorar la aplicabilidad práctica del modelo desarrollado.

8 ANEXO

8.1 Ecuaciones – Método de las Diferencias Finitas

Ecuación diferencial de la Difusión del Calor en estado estacionario

$$\frac{\partial^2 T(x, y)}{\partial x^2} + \frac{\partial^2 T(x, y)}{\partial y^2} = 0 \quad (1)$$

Se parte de la discretización del dominio de la chapa en un conjunto de puntos/nodos que esta distribuidos de forma regular.

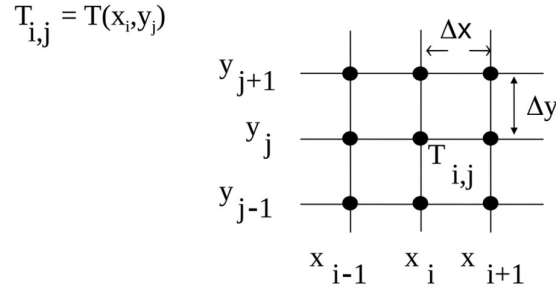


Figura 11: Discretización de la chapa

Se discretizan las derivadas segundas de (1) y se unifican los índices.

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} \quad y \quad \frac{\partial^2 T}{\partial y^2} = \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2}$$

Tomando $\beta = \frac{\Delta x}{\Delta y}$

$$T_{i-1,j} + \beta^2 T_{i,j-1} - 2(1 + \beta^2)T_{i,j} + T_{i+1,j} + \beta^2 T_{i,j+1} = 0 \quad (2)$$

Tomando $k = i + jN_x$ se obtiene:

$$T_{k-1} + \beta^2 T_{k-N_x} - 2(1 + \beta^2)T_k + T_{k+1} + \beta^2 T_{k+N_x} = 0 \quad (3)$$

La ecuación (3) es valida para todos los nodos internos de la chapa.

Aplicando las diferentes condiciones de contorno (temperatura y flujo constante) en los bordes (A, B, C y D) se llegan a las siguientes expresiones:

Borde A:

Si la temperatura es constante:

$$T_k = T_A$$

Si el flujo es constante:

$$\beta^2 T_{k-N_x} - 2(1 + \beta^2)T_k + 2T_{k+1} + \beta^2 T_{k+N_x} = 2\Delta x q_{x,A}$$

Borde B:

Si la temperatura es constante:

$$T_k = T_B$$

Si el flujo es constante:

$$\beta^2 T_{k-N_x} - 2(1 + \beta^2)T_k + 2T_{k-1} + \beta^2 T_{k+N_x} = -2\Delta x q_{x,B}$$

Borde C:

Si la temperatura es constante:

$$T_k = T_C$$

Si el flujo es constante:

$$T_{k-1} - 2(1 + \beta^2)T_k + T_{k+1} + 2\beta^2 T_{k+N_x} = 2\beta^2 \Delta x q_{y,C}$$

Borde D:

Si la temperatura es constante:

$$T_k = T_D$$

Si el flujo es constante:

$$T_{k-1} - 2(1 + \beta^2)T_k + T_{k+1} + 2\beta^2 T_{k-N_x} = -2\beta^2 \Delta x q_{y,D}$$

8.2 Registro de prueba, hiperparámetros y archivos asociados

Se presenta un resumen de las diferentes configuraciones estudiadas en este trabajo.

Tipo de entrenamiento	Prueba N°	Muestras	Hiperparámetros utilizados en el entrenamiento							Nombre de archivos asociados		Observaciones generales
			lr	Epoch	Batch Size	α	r_n	RMSE Final [°C]	Observaciones	Dataset	Modelo	
Modelo con dataset amplio – Sin ponderación	1	1000	1e-3	1000	32	-	-	337,8		dataset_1000_test	MLP_1000_test_run3	
	2	5000	1e-3	1000	32	-	-	146,3		dataset_5000_test	MLP_5000_test_run3	
	3	50 000	1e-3	1000	32	-	-	52,5		dataset_50000_test	MLP_50000_test_run3	
	4	100 000	1e-3	1000	32	-	-	29,6		dataset_100000_test	MLP_100000_test_run3	
	5	1000	1e-4	1000	32	-	-	342,7		dataset_1000_test	MLP_1000_test_run4	
	6	5000	1e-4	1000	32	-	-	203,0		dataset_5000_test	MLP_5000_test_run4	
	7	50 000	1e-4	1000	32	-	-	57,5		dataset_50000_test	MLP_50000_test_run4	
	8	100 000	1e-4	1000	32	-	-	28,0		dataset_100000_test	MLP_100000_test_run4	

	9	200 000	1e-4	1000	32	-	-	24,7		dataset_200000_test	MLP_200000_t est_run4	
	10	100 000	1e-4	2000	32	-	-	24,5		dataset_100000_test	MLP_100000_t est_run42	No se observa una disminución pronunciada de la función de pérdida -. No vale la pena aumentar las épocas bajo estas condiciones
Modelo con dataset amplio – Con ponderación	11	1000	1e-4	1000	64	0.5	2	333,9		dataset_1000_test	MLP_1000_test _run00011	
	12	5000	1e-4	1000	64	0.5	2	200,5		dataset_5000_test	MLP_5000_test _run00011	
	13	50 000	1e-4	1000	64	0.5	2	89,3		dataset_50000_test	MLP_50000_tes t_run00011	
	14	100 000	1e-4	1000	64	0.5	2	72,6		dataset_100000_test	MLP_100000_t est_run00011	
	15	200 000	1e-4	1000	64	0.5	2	69,96		dataset_200000_test	MLP_200000_t est_run00011	
	16	200 000	1e-4	1000	64	0.3	2			dataset_200000_test	MLP_200000_t est_run00013	
	17	100 000	1e-4	1000	64	0.3	1			dataset_100000_test	MLP_100000_t est_run00014	

	18	100 000	1e-3	1000	64	0.3	1	69.9		dataset_100000_test	MLP_100000_t est_run00015	
	19	100 000	1e-3	1000	64	0.2	1	62.4		dataset_100000_test	MLP_100000_t est_run00016	
	20	100 000	1e-3	1000	64	0.1	1	47.3		dataset_100000_test	MLP_100000_t est_run00017	
	21	100 000	1e-3	1000	64	0.05	1	38.8		dataset_100000_test	MLP_100000_t est_run00018	
	22	100 000	1e-3	1000	64	0.025	1	34.1		dataset_100000_test	MLP_100000_t est_run00019	
	23	200 000	1e-3	1000	64	0.025	1			dataset_200000_test	MLP_200000_t est_run00019	
	24	100 000	1e-3	1000	64	0.025	1		Modificación en la ventana de selección – se consideró un círculo	dataset_100000_test	MLP_100000_t est_run00019R	No se observó mejora en esta configuración. El loss del radio aumentó.
Modelo con dataset restringido	25	100 000	1e-3	1000	64	-	-		La curva de validación presenta oscilaciones marcadas	dataset_100000_test _old	MLP_100000_t est_old_run03	Convergencia en 250 epoch
	26	100 000	1e-4	1000	64	-	-		La curva de validación presenta oscilaciones	dataset_100000_test _old	MLP_100000_t est_old_run04	Convergencia en 700 epoch

