

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE
TECNOLOGÍAS Y SERVICIOS DE
TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**IMPLEMENTACIÓN DEL MODELADO DE
TEMPERATURA MEDIANTE ALGORITMOS
DE APRENDIZAJE AUTOMÁTICO PARA
SISTEMAS DE REFRIGERACIÓN POR
INMERSIÓN EN HIDRO-FLUORO-ÉTERES**

JAIME PÉREZ SÁNCHEZ

2018

TRABAJO FIN DE GRADO

Título: Implementación del modelado de temperatura mediante algoritmos de aprendizaje automático de sistemas de refrigeración por inmersión en Hidro-Fluoro-Éteres

Autor: Jaime Pérez Sánchez

Tutor: Patricia Arroba García

Departamento: Ingeniería Electrónica

MIEMBROS DEL TRIBUNAL

Presidente:

Vocal:

Secretario:

Suplente:

FECHA DE LECTURA:

CALIFICACIÓN:

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



**GRADO EN INGENIERÍA DE TECNOLOGÍAS Y
SERVICIOS DE TELECOMUNICACIÓN**

TRABAJO FIN DE GRADO

**IMPLEMENTACIÓN DEL MODELADO DE
TEMPERATURA MEDIANTE ALGORITMOS
DE APRENDIZAJE AUTOMÁTICO PARA
SISTEMAS DE REFRIGERACIÓN POR
INMERSIÓN EN HIDRO-FLUORO-ÉTERES**

JAIME PÉREZ SÁNCHEZ

2018

Resumen

Uno de los problemas más críticos en los centros de procesamiento de datos es la refrigeración. Las técnicas de refrigeración actuales son poco eficientes tanto en términos de energía, consumiendo hasta un 40% del total del centro de datos, como en área ocupada. Esto supone un problema crítico para el desarrollo de las nuevas ciudades inteligentes, que requieren el despliegue de numerosos centros de datos en núcleos urbanos que procesen aplicaciones de Data Analytics en tiempo real.

En este trabajo se ha desarrollado una nueva solución disruptiva para este problema, que consiste en la inmersión de la infraestructura de computación en un líquido dieléctrico basado en hidro-fluoro-éteres (HFE). De esta forma se consigue una refrigeración pasiva de dos fases eliminando el consumo energético de refrigeración. El líquido tiene buenas propiedades térmicas, es un buen aislante eléctrico y además es respetuoso con el medio ambiente. La capacidad de arrastre de calor del HFE es mucho mayor que la del aire, lo que hace posible aumentar la densidad de computación de los centros de datos reduciendo su área. No obstante, para asegurar la capacidad máxima de arrastre de calor del HFE, es necesario asegurar unas condiciones térmicas específicas.

Realizar un modelo predictivo es crucial para cualquier sistema que desee trabajar alrededor del punto de máxima eficiencia. Por lo tanto, este proyecto se centra en el diseño y la implementación de un modelo predictivo de temperatura que permita tomar decisiones acerca de la carga de trabajo introducida en el centro de datos, con un margen de tiempo suficientemente amplio para mantener el sistema trabajando a ciertas temperaturas de una manera óptima.

En este proyecto se ha obtenido con éxito un modelo térmico predictivo basado en una arquitectura de red neuronal artificial recurrente, con neuronas tipo GRU (Gated Recurrent Unit). Dicho modelo realiza satisfactoriamente predicciones acerca de la temperatura de un sistema basado en refrigeración por inmersión en fluido HFE con una ventana temporal de 1 minuto y una media de error de 0.753 °C.

Palabras clave

Centros de datos, ciudades inteligentes, refrigeración por inmersión, modelado de temperatura, aprendizaje automático, redes neuronales.

Summary

Optimizing the cooling infrastructure is one of the main challenges in data center's scope. Current cooling techniques are not very efficient both in terms of energy, consuming up to 40% of the total energy requirements, and in occupied area. This is a critical problem for the development of new smart cities, which require the deployment of numerous data centers in urban areas to process Data Analytics applications in real time.

In this work a new disruptive solution has been developed to address this problem, which consists of the immersion of the computing infrastructure in a dielectric liquid based on hydro-fluoro-ethers (HFE). In this way, a passive cooling of two phases is achieved, eliminating the energy consumption of cooling. The liquid has good thermal properties, it is a good electrical insulator and it is also respectful with the environment. The heat transfer capacity of the HFE is higher than in air-based systems, which makes it possible to increase the computing density of data centers by reducing their area. However, to ensure the maximum heat transfer capacity of the HFE, it is necessary to ensure specific thermal conditions.

Making a predictive model is crucial for any system that needs to work around the point of maximum efficiency. Therefore, this project focuses on the design and implementation of a predictive temperature model that allows decisions to be made about the workload allocation in the data center, with enough time to keep the system working at certain temperatures in an optimal way.

In this manuscript, we successfully obtained a predictive thermal model using a GRU (Gated Recurrent Unit)-based artificial neural network architecture. This model makes accurate thermal predictions of a system based on HFE immersion cooling that presents an average error of 0.753°C with a time window of 1 minute.

Keywords

Data centers, smart cities, immersion cooling, thermal modeling, machine learning, neural networks.

Índice de contenido

1. Introducción	1
1.1 Las ciudades del futuro	1
1.2 El papel de la tecnología	2
1.3 Retos derivados del uso de tecnología	3
1.4 El reto más importante: La energía	4
1.5 La climatización en centros de datos	6
1.6 Solución: Inmersión pasiva bi-fase	7
1.7 Fases del proyecto	9
2. Estado del arte	10
2.1 Refrigeración en centros de datos	10
2.1.1 Refrigeración por corrientes de aire	10
2.1.2 Refrigeración por agua	12
2.1.3 Refrigeración por inmersión	14
2.2 Modelos predictivos de temperatura	18
2.2.1 Modelos analíticos	18
2.2.2 Modelos metaheurísticos	18
2.2.3 Aprendizaje automático y profundo	19
3. Solución	24
3.1 Montaje físico del sistema	24
3.2 Diseño de la carga de trabajo	26
3.3 Software de captura y visualización	28
3.4 Modelo predictivo de temperatura	29
3.4.1 Herramientas de software utilizadas	29
3.4.2 Hiperparámetros	30
3.4.3 Métricas de evaluación de modelos predictivos	31
4. Experimentos	32
4.1 Modelo base	33
4.2 Red basada en <i>SimpleRNN</i>	35
4.2.1 Estructura neuronal	35
4.2.2 Función de activación	35
4.2.3 <i>Batch Size</i>	36
4.2.4 <i>Epochs</i>	36

4.3 Red basada en LSTM	38
4.3.1 Estructura neuronal	38
4.3.2 Función de activación	38
4.3.3 <i>Batch Size</i>	39
4.3.4 <i>Epochs</i>	40
4.4 Red basada en GRU	41
4.4.1 Estructura neuronal	41
4.4.2 Función de activación	42
4.4.3 <i>Batch Size</i>	43
4.4.4 <i>Epochs</i>	43
5. Resultados	45
5.1 Comparación de modelos	45
5.2 Modelo seleccionado	46
6. Conclusiones y líneas futuras	49
7. Bibliografía	51
Anexo A: Aspectos éticos, económicos, sociales y ambientales	54
A.1 Introducción	54
A.2 Descripción de impactos relevantes relacionados con el proyecto	54
A.3 Análisis detallado de alguno de los principales impactos	55
A.4 Conclusiones	55
Anexo B: PRESUPUESTO ECONÓMICO	56

Índice de figuras

Figura 1: Previsión del porcentaje de población urbana en 2050, ONU	1
Figura 2: Evolución número de dispositivos IoT en miles de millones, IHS <i>Markit</i>	2
Figura 3: Diagrama jerárquico de la computación <i>Cloud y Edge</i>	4
Figura 4: Desglose del consumo energético en un Centro de Procesamiento de Datos	5
Figura 5: Sistema de refrigeración basado en CRAC	6
Figura 6: Capacidad teórica de arrastre de calor en función de la temperatura, 3M	8
Figura 7: Diagrama de ejemplo de una red neuronal artificial de 4 capas	9
Figura 8: Sistema de refrigeración basado en CRAC	10
Figura 9: Sistema de refrigeración CRAC con aislamiento de pasillo caliente	11
Figura 10: Sistema de refrigeración <i>free-cooling</i>	11
Figura 11: Sistema de refrigeración <i>water-cooling</i> de puertas traseras activas	13
Figura 12: Sistema de refrigeración CRAH	13
Figura 13: Sistema de refrigeración por inmersión de una fase	15
Figura 14: Sistema de refrigeración por inmersión pasiva de dos fases	15
Figura 15: Sistema de refrigeración <i>Carnotjet</i> en el supercomputador <i>Tsubame KFC</i>	16
Figura 16: Sistema refrigeración por inmersión de la empresa <i>Iceotope</i>	16
Figura 17: Sistema <i>BlockBox IC</i> de minado de <i>Bitcoins</i>	17
Figura 18: Sistema <i>GreenICE</i> de redes de datos de alto rendimiento	17
Figura 19: Representación del problema de compromiso entre búsqueda diversificada e intensiva	19
Figura 20: Esquema de una red neuronal convolucional	21
Figura 21: Esquema de una neurona <i>SimpleRNN</i> desenrollada en la secuencia temporal	21
Figura 22: Esquema de una neurona LSTM desenrollada en la secuencia temporal	22
Figura 23: Esquema de una neurona GRU	22
Figura 24: Esquema de la solución	24
Figura 25: Placa <i>Raspberry Pi 3 Model B+</i> y cluster de 3 placas	24
Figura 26: Fotografías del prototipo	25
Figura 27: Porcentaje de utilización de CPU en cada ejecución cíclica	28
Figura 28: Servidor web de <i>GreenLSI</i> con la herramienta Graphite	29

Figura 29: Esquema de los experimentos	32
Figura 30: Predicción de temperatura del modelo inicial basado en LSTM	33
Figura 31: Predicción de red basada en <i>SimpleRNN</i> tras optimizar hiperparámetros	37
Figura 32: Predicción de red basada en LSTM tras optimizar hiperparámetros	41
Figura 33: Predicción de red basada en GRU tras optimizar hiperparámetros	44
Figura 34: Comparación de predicciones de los tres tipos de redes neuronales	45
Figura 35: Predicción de la red neuronal basada en GRU con ventana temporal de 1 minuto	46
Figura 36: Histograma de errores en la predicción de la red neuronal basada en GRU	46
Figura 37: Comparación de predicciones con distintas ventanas temporales	47
Figura 38: Evolución del error medio absoluto y desviación estándar (MAE \pm STD) en función de la ventana de predicción	48
Figura 39: Comparación de predicciones con distintas ventanas temporales	48

1. Introducción

1.1 Las ciudades del futuro

Las ciudades han reemplazado poco a poco a las zonas rurales como nuestro hábitat natural. Según datos de la Organización de las Naciones Unidas, en 2008 por primera vez en la historia de la humanidad vivía más gente en núcleos urbanos que en zonas rurales. Esta tendencia a la concentración poblacional ha seguido aumentando y las proyecciones para el futuro indican que será un factor cada vez más importante.

En 2015, un informe de la ONU [1] sobre distribución de población mundial estimaba que un 54% de los habitantes del planeta ya vivía en una ciudad, y que en 2050 esta cifra llegaría hasta el 66%. En Europa, uno de los continentes más urbanizados, el 73% de la población vivía en ciudades y se estimaba que en 2020 llegaría al 80%.

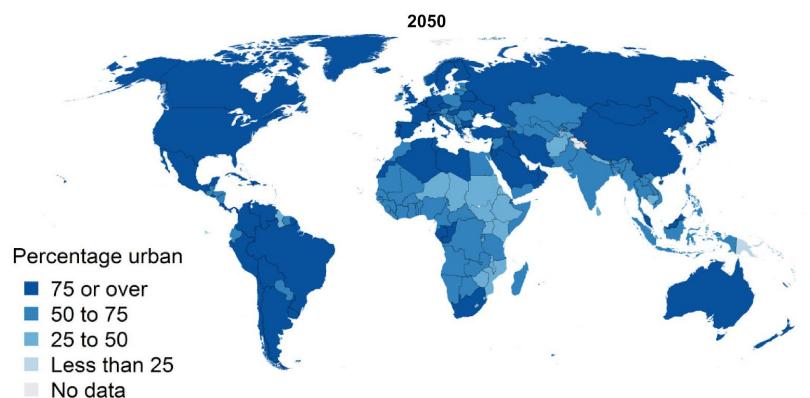


Figura 1: Previsión del porcentaje de población urbana en 2050, ONU

Estos datos son más preocupantes aún en términos de salud pública si tenemos en cuenta que las ciudades son los mayores focos de contaminación del planeta. Un estudio de la Universidad de Carolina del Norte [2] publicado en *Nature's Climate Change*, pronostica que si continúa la tendencia actual, la contaminación atmosférica será responsable de 60.000 muertes prematuras en el año 2030 y de 260.000 en 2100. También debemos tener en cuenta que, según los últimos informes de la *International Energy Agency* (IEA) [3], alrededor del 70% de la energía mundial se consume en las ciudades. Además, se espera que la demanda de energía crezca un 40% en el año 2030 con respecto a la actual. [4]

Todos estos datos apuntan a la necesidad de crear ciudades sostenibles, eficientes energéticamente y respetuosas con el medio ambiente, que garanticen espacios de bienestar y calidad de vida a sus ciudadanos. Estas ciudades del futuro se conocen como ciudades inteligentes o ***Smart Cities***.

1.2 El papel de la tecnología

El imparable proceso de digitalización y conexión hace que el pilar más importante en el que se sostienen estas ciudades sea la tecnología, especialmente lo que se conoce como **Internet of Things (IoT)**.

El *Internet of Things* consiste en la interconexión digital de objetos cotidianos con Internet. Esto nos permite que los objetos se comuniquen entre sí y con centros de control, para que puedan actuar de manera más inteligente frente a diferentes estímulos o situaciones. Nos da acceso a una gran cantidad de información de nuestro entorno y además crea nuevas líneas de ingresos y fórmulas de consumo diferentes.

Las ventajas de su uso son claras: mejoras en la eficiencia operativa y productiva, mayor facilidad de monitorización y actuación, aumento de los beneficios económicos y disminución de la intervención humana, que es más propensa a fallos.

Las posibles aplicaciones son muy variadas: los denominados *wearables* (como *smartwatch*, zapatillas, tejidos inteligentes, etc), la domótica (neveras inteligentes, calefacción, riego, etc), iluminación de calles inteligente, transporte público inteligente, educación interactiva, *E-Health*, seguridad, vigilancia y avisos de contaminación en tiempo real entre otros.

Hoy en día hay alrededor de 23.000 millones de dispositivos de IoT conectados. Pero se calcula que en 2030 habrá más de 100.000 millones. [5]

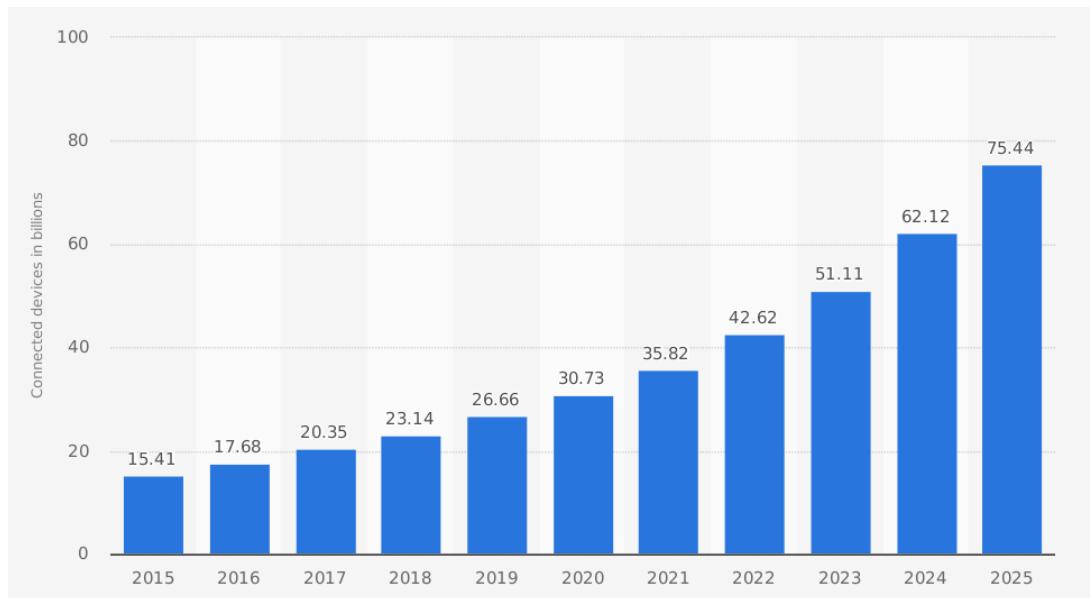


Figura 2: Evolución número de dispositivos IoT en miles de millones, IHS Markit

Las Smart Cities tendrán una gran cantidad de sensores repartidos por la ciudad, y por lo tanto una enorme cantidad de información (**Big Data**). Esta se puede utilizar para

ayudar en la toma de decisiones tanto a largo plazo, como en tiempo real para algunas aplicaciones.

Para ello necesitamos aplicar sobre estos datos algoritmos metaheurísticos que aprendan patrones y relaciones entre las distintas fuentes de información. Por ejemplo los basados en redes neuronales artificiales (*Artificial Neural Networks* o ANN) como el aprendizaje automático (*Machine Learning*), en aprendizaje profundo (*Deep Learning*). El uso de estos algoritmos sobre grandes cantidades de datos se conoce como ***Data Analytics***.

Hay gran cantidad de aplicaciones que se pueden implementar utilizando *Data Analytics*. Por ejemplo, evitar atascos con gestión automática de semáforos y vías de emergencia, *Smart Parking*, predicciones de contaminación, gestión de residuos, detección de fugas o anomalías en suministros, seguridad con reconocimiento facial o de voz, predicción de picos de consumo energético y prevención de accidentes de tráfico entre otras aplicaciones.

1.3 Retos derivados del uso de tecnología

Actualmente hay un gran problema con el análisis masivo de datos debido a que para utilizar estos algoritmos se necesita una gran potencia de cómputo que típicamente se realiza en la nube (***Cloud Computing***). Además, toda esa información se debe enviar previamente a los servidores, lo cual aumentará enormemente el tráfico de la red.

Por otra parte, la mayoría de los dispositivos de IoT no realizan gran carga de cómputo, si no que también envían los datos a los servidores de la nube y esta se encarga de realizar los cálculos. Algunos sistemas conectados requieren por sí solos una gran capacidad de cálculo y de transmisión de datos. Un claro ejemplo de esto son los sistemas de asistencia a la conducción (*Advanced Driver Assistance Systems* o ADAS). Se estima que pueden generar 4.000GB de datos por vehículo cada día, con una computación de pico asociada de 1GB/s. [6]

Teniendo en cuenta estos factores y el número de dispositivos IoT conectados que habrá en pocos años, corremos un gran riesgo de saturar los centros de datos (*Data Center*), de incrementar el coste de la computación y de aumentar demasiado las latencias para obtener respuestas en tiempo real.

Para solucionar este problema, en los últimos años se ha propuesto una solución conocida como computación en el borde o ***Edge Computing***. Consiste en la utilización de dispositivos más cercanos que el Cloud a las fuentes de generación de datos (***Edge Data Centers***), para que realicen la mayor parte de la carga computacional, de almacenamiento y de comunicación.

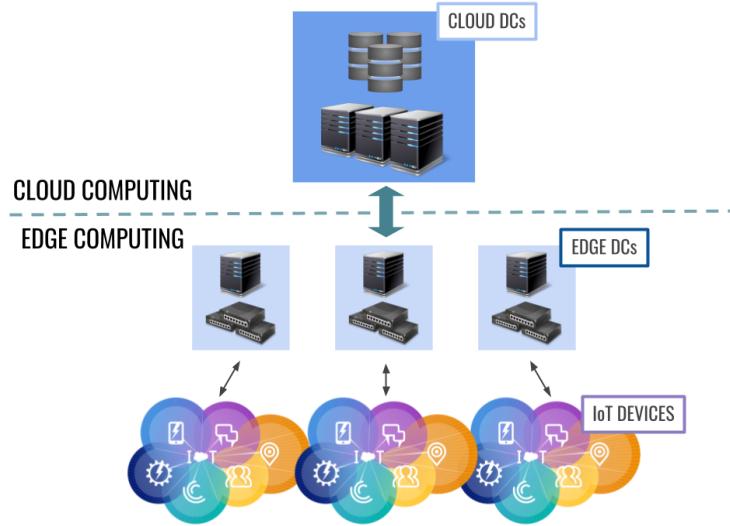


Figura 3: Diagrama jerárquico de la computación *Cloud* y *Edge*

La utilización de esta arquitectura tiene grandes ventajas: evitamos la saturación de los centros de datos, conseguimos un descongestionamiento de la red y disminuimos la latencia asociada al *Cloud Computing*.

1.4 El reto más importante: La energía

Sin embargo, esta solución viene asociada con otro importante problema de los *Data Centers*, que es el consumo de energía. Hoy en día la industria de los centros de datos consume más del 2% de la producción mundial de energía [7]. El *Data Center* más potente del mundo consume alrededor de 200.000 MWh anuales y se calcula que en 2020 el que ostente este puesto, consumirá el equivalente a la producción de una central nuclear de tamaño medio [8]. Actualmente, la suma de los costes energéticos de los más de 500.000 centros de datos repartidos por el mundo, alcanza la abrumadora cifra de 200.000 millones de euros al año.

Las fuentes de consumo más importantes en los centros de datos son las infraestructuras de computación y refrigeración, necesaria para evitar caídas del sistema y fallos irreparables por alcanzar temperaturas críticas en el hardware. Los recursos de computación representan alrededor de un 50% del consumo total y la climatización supone un 34% del total [9]. El resto del consumo energético viene dado por las pérdidas de energía en los transformadores, el Sistema de Alimentación Interrumpida (SAI), la iluminación, y otros elementos.

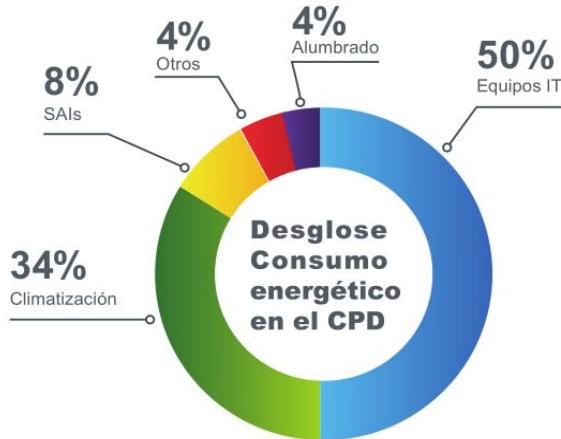


Figura 4: Desglose del consumo energético en un Centro de Procesamiento de Datos

Parece razonable considerar la climatización como el factor más importante al que atacar en cuestión de eficiencia energética. Una de las métricas más aceptadas en los *Data Centers* en cuestión de eficiencia energética es el **PUE (Power Usage Effectiveness)**. Fue desarrollado por el consorcio industrial *The Green Grid*, que se dedica a mejorar la eficiencia de los recursos en los centros de datos. El PUE es un ratio entre el consumo total del centro de datos en un año, dividido por el consumo propio de los equipos informáticos ese mismo año. Es decir, nos da la medida de cuántos kWh de consumo total tenemos por cada kWh consumido por los equipos informáticos.

$$PUE = \frac{\text{Energía Total DC}}{\text{Energía Equipos IT}} = 1 + \frac{\text{Energía Equipos No IT}}{\text{Energía Equipos IT}}$$

Si un *Data Center* presenta un PUE igual a 2, significa que el consumo de la infraestructura completa es 2 veces mayor al consumo de los equipos informáticos. Por lo tanto, cuanto menor es el PUE, más eficiente es el centro de datos. Siendo 1 el PUE ideal, es decir, cuando el consumo total del *Data Center* se deba únicamente al consumo eléctrico de los equipos informáticos.

Los macrocentros de datos de *Google* tienen un PUE de 1,11 [10] y los de *Facebook* 1,07 [11], pero la media mundial se sitúa en 1,79. Los PUEs más bajos se consiguen en muchos casos situando el centro de datos en zonas geográficas lo más frías posibles. No obstante, es evidente que esta solución no se puede aplicar al *Edge Computing*, ya que su ubicación viene fijada por la fuente de origen de datos. Por tanto, uno de los principales retos para *Edge Data Centers* es el diseño de sistemas de refrigeración eficientes, independientemente del clima.

1.5 La climatización en centros de datos

El método de enfriamiento más utilizado hoy en día en centros de datos es la refrigeración por corrientes de aire. Estas corrientes se controlan mediante las unidades CRAC (*Computer Room Air Conditioner*) que las conduce por un suelo técnico hacia los racks de servidores, con configuraciones especiales de pasillo frío - pasillo caliente.

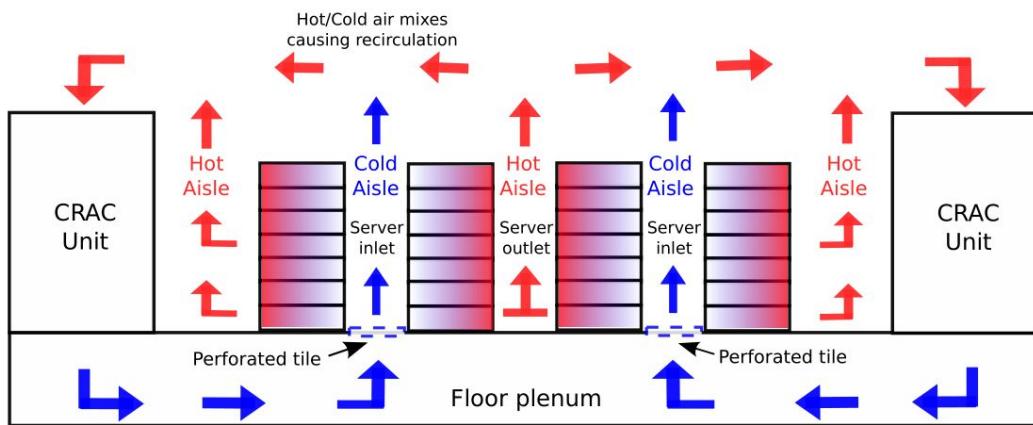


Figura 5: Sistema de refrigeración basado en CRAC

Estos sistemas de refrigeración se manifiestan altamente ineficientes e insuficientes para las necesidades a medio y largo plazo. Esto es debido, entre otras razones, a su escasa adaptación a las condiciones específicas de cada sala, su gran consumo energético y su elevado coste.

Existen distintas propuestas que mejoran algunos de los aspectos negativos, como el aislamiento de los pasillos calientes o los pasillos fríos, pero sigue resultando una solución demasiado costosa y compleja para un *Edge Data Center*. La refrigeración de los *Data Centers* es uno de los principales cuellos de botella de cara a su evolución.

En el apartado acerca del estado del arte, se explicarán más detalladamente los tipos de refrigeraciones existentes hoy en día, y las razones por las que no resultan una buena opción para los *Edge Data Center*.

Si deseamos desplegar pequeños centros de datos en el interior de las ciudades y así asegurar un desarrollo sostenible de las nuevas tecnologías inteligentes, estos deben ocupar el menor espacio posible por factores de comodidad para los ciudadanos y coste de los espacios de despliegue. Además deben reducir al mínimo el consumo de la climatización, ya que la proliferación de estas infraestructuras amenaza con saturar las redes eléctricas de las ciudades, provocando cortes en el servicio que tendría un alto impacto económico.

1.6 Solución: Inmersión pasiva bi-fase

En este proyecto estudiaremos una solución completa para la refrigeración de los *Edge Data Centers* basada en la inmersión de las placas base de los servidores en un líquido dieléctrico, el *Novec 7100* de la empresa *3M* [12], para realizar una refrigeración pasiva de dos fases.

Este líquido formado por **Hidro-Fluoro-Éteres (HFE)**, es un disolvente orgánico complejo que se presenta en forma de mezclas de isómeros inseparables (metoxi-heptafluorobutano). Fue desarrollado como reemplazo para compuestos químicos utilizados principalmente en industria como CFC, HFC, HCFC y PFC, que son perjudiciales para la capa de ozono y favorecen el efecto invernadero. El HFE en cambio, es respetuoso con el medio ambiente y de muy baja toxicidad.

Inicialmente se utilizó para la limpieza de equipos electrónicos, y más tarde en la extinción de incendios en situaciones en las que el agua dañaría activos de valor y para la preservación de componentes biológicos entre otros usos.

Se caracteriza por ser incoloro, inodoro, no inflamable, de baja toxicidad, baja tensión superficial, baja viscosidad y se encuentra en estado líquido a temperatura ambiente. Visualmente es indistinguible del agua, pero su punto de ebullición es de 61°C y su densidad es de 1,52 g/ml [12].

Este conjunto de características permiten una manipulación segura y de bajo impacto ambiental. El fabricante pronostica diversas ventajas derivadas de su implementación [13], que soluciona los problemas de *Edge Computing* mencionados anteriormente:

- Al ser una refrigeración pasiva, reducirá un 95% los costes energéticos de climatización.
- Aumentará de la densidad de potencia por rack de 40 kW a 250 kW.
- Cada *Data Center* utilizará 10 veces menos espacio que en la actualidad.
- Se podrán alcanzar PUEs por debajo de 1,02.
- Se simplificará la construcción e instalación de los *Data Centers*.
- Se reducirá el mantenimiento necesario al reducir en número de partes móviles y otros componentes.
- Se reducirán los costes de los servidores al no necesitar cajas, ventiladores...
- Mejorará significativamente el tiempo medio entre fallos, por la reducción de la media de la temperatura y las variaciones drásticas de esta.

Nuestro grupo de investigación, *GreenLSI*, está trabajando actualmente en descubrir y explorar los límites de un sistema de refrigeración basado en esta tecnología. Especialmente, el trabajo se centra en conocer la capacidad de arrastre de calor máxima y la temperatura de

trabajo más eficiente del líquido refrigerante. Con ello se espera obtener una solución completa para el despliegue de *Edge Data Centers*.

Según los cálculos teóricos realizados por el fabricante, la capacidad de arrastre de calor (W/cm^2) del fluido en función de la temperatura seguiría la forma de la siguiente figura:

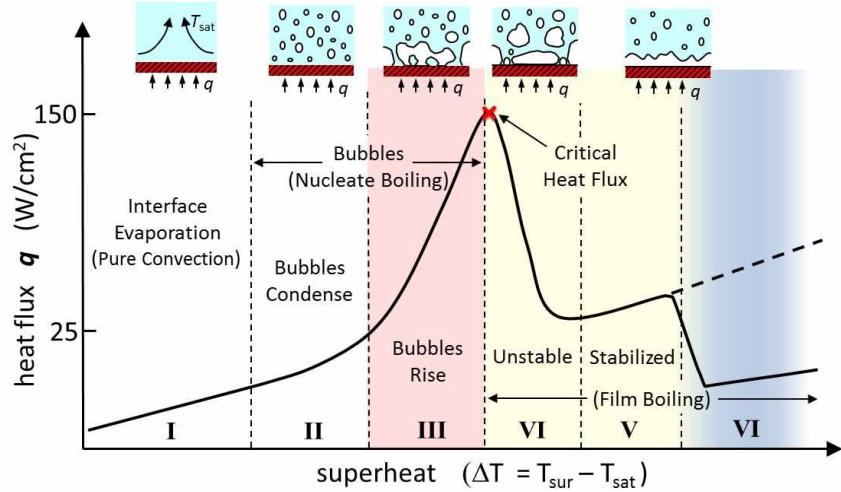


Figura 6: Capacidad teórica de arrastre de calor en función de la temperatura, 3M

Como podemos observar, existe cierto margen de temperaturas (III) en el que el arrastre de calor del fluido es muy elevado, pero si la temperatura sube demasiado (IV), esta capacidad cae de forma drástica. Si deseamos que un sistema que utilice esta tecnología trabaje de la forma más eficiente posible, debemos mantener su temperatura en un punto cercano al máximo.

Por lo tanto, este proyecto se centra en el diseño y la implementación de un modelo predictivo de temperatura que permita tomar decisiones acerca de la carga de trabajo introducida, con un margen de tiempo suficientemente amplio para mantener el sistema trabajando a ciertas temperaturas de una manera óptima.

Realizar un modelo predictivo es crucial para cualquier sistema que desee trabajar alrededor del punto de máxima eficiencia. En el caso concreto de la temperatura en sistemas electrónicos que utilicen técnicas de refrigeración, es más importante si cabe ya que esta variable es un factor determinante para que los equipos y componentes trabajen de manera correcta y rápida.

La ventaja más destacable de utilizar la temperatura como variable a modelar, es que esta tiene una importante inercia. Es decir, la temperatura no sufre cambios demasiado bruscos y está directamente relacionada con otras variables que somos capaces de obtener fácilmente como el porcentaje de utilización de CPU, la frecuencia de CPU, la evolución de la temperatura del propio equipo minutos antes y la temperatura de los dispositivos cercanos entre otras. Una vez desarrollado el modelo de temperatura, podremos anticipar acciones directas o indirectas de optimización, y así mantener el sistema en su punto de máxima eficiencia.

Para la realización de dicho modelo predictivo utilizaremos algoritmos de aprendizaje automático (**Machine Learning**) y aprendizaje profundo (**Deep Learning**), basados en capas de redes neuronales artificiales (**Artificial Neural Networks** o ANN). Estos algoritmos, que tratan de imitar el funcionamiento de las redes neuronales humanas, forman parte de una rama de la inteligencia artificial y las ciencias de la computación. Su objetivo es desarrollar técnicas que permitan a los ordenadores aprender a descifrar patrones y correlaciones, mediante el entrenamiento con grandes cantidades de datos.

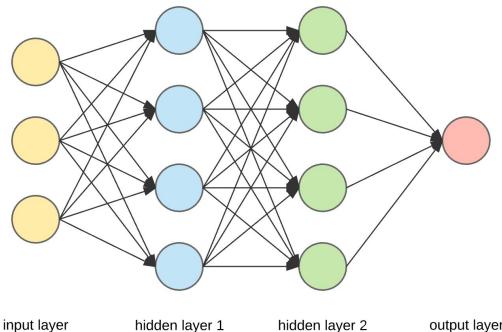


Figura 7: Diagrama de ejemplo de una red neuronal artificial de 4 capas

La elección de estos algoritmos para el desarrollo del modelo predictivo de temperatura se desarrolla más detalladamente en el estado del arte. Pero se basa principalmente en la dificultad técnica de realizar modelos analíticos para la predicción de sistemas complejos y la facilidad de implementación que ofrecen diversas librerías de código abierto como **Keras** [14] y **Tensor Flow** [15], esta última desarrollada por *Google*. Además de la multitud de ejemplos y aplicaciones en las que estos algoritmos resuelven satisfactoriamente diversos problemas de predicción. [16]

Para el despliegue físico de la solución presentada en el este proyecto, se hará uso de un contenedor de barco en el que se instalará un cluster con varias *Raspberry Pi 3 Model B+* y se sumergirá en el líquido dieléctrico. Se utilizará una carga de trabajo de Data Analytics para simular el entorno real de *Edge Data Center* en una *Smart City*.

1.7 Fases del proyecto

1. Montaje físico del sistema basado en placas *Raspberry Pi 3 Model B+*.
2. Elección de software de captura de datos relevantes, así como una plataforma de visualización y almacenamiento.
3. Diseño de la carga de trabajo basada en *Data Analytics* simulando un entorno real de *Edge Data Center* en *Smart City* e implementación de la captura de datos.
4. Diseño e implementación de un modelo predictivo de temperatura utilizando algoritmos de aprendizaje automático y profundo (*Machine* y *Deep Learning*).
5. Obtención de conclusiones y resultados acerca del modelo predictivo de temperatura y la viabilidad general del sistema.

2. Estado del arte

En este apartado se presenta el estudio de las soluciones que se ofrecen y utilizan en la actualidad para resolver los problemas de refrigeración en centros de datos y la realización de modelos predictivos de temperatura.

2.1 Refrigeración en centros de datos

2.1.1 Refrigeración por corrientes de aire

Como se ha explicado en el apartado de introducción, tradicionalmente los *Data Centers* han utilizado refrigeración basada en circulación por corrientes de aire. Sin embargo, estos sistemas se manifiestan claramente ineficientes para las necesidades futuras y son propensos a la aparición de corrientes de aire turbulentas y puntos calientes.

El método más utilizado para controlar las corrientes de aire utiliza unidades **CRAC** (*Computer Room Air Conditioner*) para conducirlas por un suelo técnico hacia los racks de servidores, con configuraciones especiales de pasillo frío - pasillo caliente.

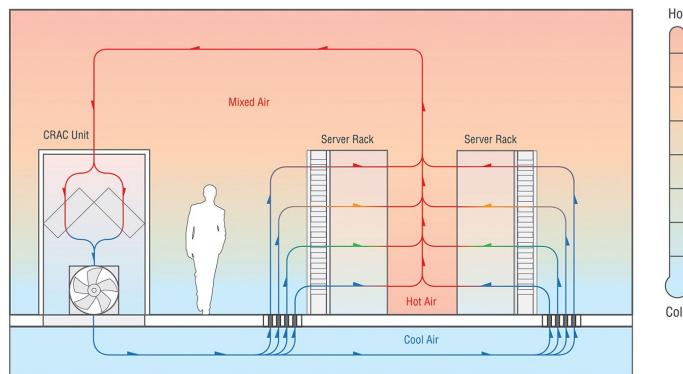


Figura 8: Sistema de refrigeración basado en CRAC

Este tipo de climatización es altamente ineficiente y costoso. Existen propuestas que mejoran algunos de los aspectos negativos de esta climatización, como el aislamiento de los pasillos calientes (ver Figura 9) o los pasillos fríos, o la utilización de rejillas con ventiladores de apoyo frente a los racks. No obstante, a medida que aumenta la densidad de potencia en los racks, este tipo de refrigeración se manifiesta menos eficiente.

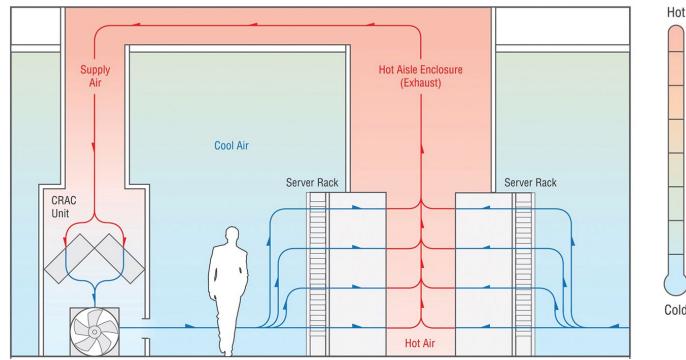


Figura 9: Sistema de refrigeración CRAC con aislamiento de pasillo caliente

Otra técnica de enfriamiento por corrientes de aire usada en *Data Centers* es la utilización de aire del exterior del recinto. Esta técnica se conoce como ventilación natural mecánica o ***free-cooling*** y, como es lógico, sólo es útil si el aire del exterior se encuentra a una temperatura menor que la del interior del centro de datos. Por lo tanto, su utilización está completamente determinada por el clima y la localización del *Data Center*.

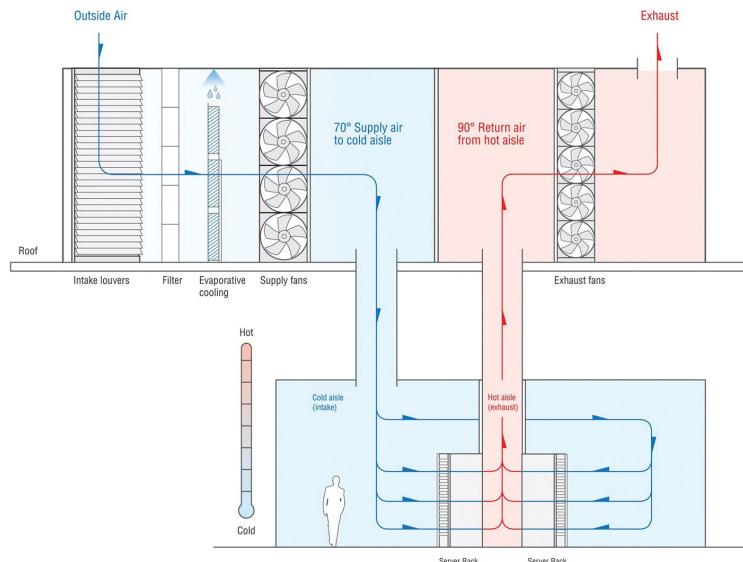


Figura 10: Sistema de refrigeración *free-cooling*

Se debe tener en cuenta que la humedad ambiental es un factor importante en un centro de datos. Si la humedad es demasiado alta podría provocar fallos irreparables en algunos componentes electrónicos, por lo que no podemos introducir aire directamente del exterior. Además, podríamos introducir polvo y otras partículas que a la larga podrían afectar al funcionamiento de los equipos.

Por lo tanto, es necesaria la instalación de deshumidificadores y filtros de aire, que aseguren que el aire introducido en las salas es adecuado en temperatura, humedad y nivel de partículas. Esto conlleva unos gastos energéticos significativos, además de un elevado coste de instalación.

Existen también ciertas soluciones más cercanas al rack conocidas como *in-rack*, basadas en dispositivos modulares que extraen el calor del rack sin dejar que este inunde la sala de servidores. Sin embargo, al igual que el resto de soluciones, no permite una gran densidad de potencia y puede dar lugar a configuraciones demasiado complejas de implementar y mantener.

En definitiva, la refrigeración por corrientes de aire tiene multitud de problemas asociados que la hacen ser un tipo de climatización destinada a desaparecer en el futuro:

- Tiene un gran consumo eléctrico pudiendo llegar al 50% en algunos casos.
- La temperatura de la sala se fija teniendo en cuenta la temperatura del servidor más caliente, ya que es imposible generar corrientes de climatización distintas para cada servidor. Por lo tanto, los servidores que tengan baja carga de trabajo o incluso se encuentren desactivados se refrigerarán más de lo necesario.
- Tiene limitaciones físicas de la cantidad de calor que puede disipar en cada servidor, lo que limita la densidad de potencia en los racks para no provocar sobrecalentamientos (los cuales producirán fallos y caídas de servicio).
- Es propenso a la creación de corrientes turbulentas de aire y puntos calientes, que disminuyen aún más su eficiencia.
- Se debe diseñar un sistema de refrigeración adaptado a la topología de cada sala, la ubicación de los servidores y la disipación de temperatura esperada en cada uno. Para ello se utilizan costosas y complejas simulaciones computacionales de dinámica de fluidos (CFD) que determinan las necesidades de climatización para una configuración concreta. Que deberá ser repetida en caso de variaciones en la distribución de la sala.
- El coste de fabricación e instalación de la infraestructura es muy elevado y supone un importante porcentaje de los costes totales de operación.
- La sala necesita un área grande, por lo el tamaño total del centro de datos aumentará considerablemente.

2.1.2 Refrigeración por agua

En algunos centros de procesamiento de datos se han introducido sistemas de refrigeración por agua o **water-cooling**, que presentan una tasa de transferencia de calor unas 23 veces mayor que la del aire [17]. No obstante, a muchos operadores les preocupa poner en riesgo los equipos electrónicos, por posibles fugas o fallos del sistema.

Una implementación bastante común de esta tecnología son las puertas traseras activas. Estas incluyen un intercambiador aire-agua con agua a bajas temperaturas, ventiladores y válvula de control de agua. Son dispositivos modulares y dinámicos capaces de proporcionar el flujo de aire y agua necesarios en cada momento. Hoy en día se pueden alcanzar densidades de potencia alrededor de 45 kW por rack [18], pero extienden la dimensión de los racks en más de 20 cm cada uno.

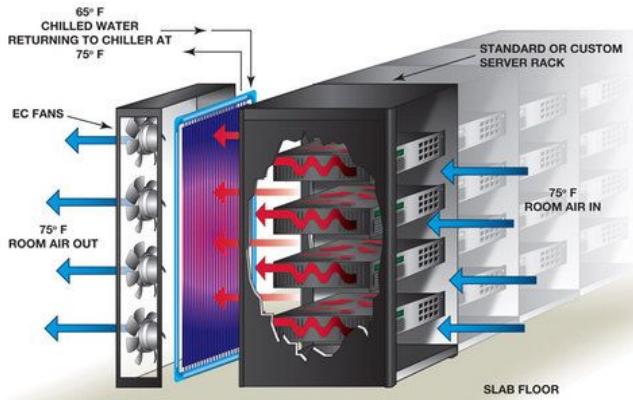


Figura 11: Sistema de refrigeración *water-cooling* de puertas traseras activas

Otra implementación se basa en dispositivos “en fila” o *in-row* que mantienen el agua a una temperatura adecuada y en constante circulación. Esto requiere el aislamiento del pasillo frío o pasillo caliente para trabajar a su máximo potencial. Se pueden alcanzar densidades de hasta 15 kW por rack, o 30 kW para soluciones muy optimizadas [19]. Pero el dispositivo ocupa mucho más espacio incluso que las puertas traseras activas.

También existen ciertas soluciones que mezclan el uso de climatización por corrientes de aire con el *water-cooling*. Para ello se utilizan unidades **CRAH** (**Computer Room Air Handler**), que funcionan de la misma manera que las unidades CRAC pero utilizando agua a baja temperatura para enfriar el aire. Estas tienen problemas similares a los comentados anteriormente acerca de las unidades CRAC.

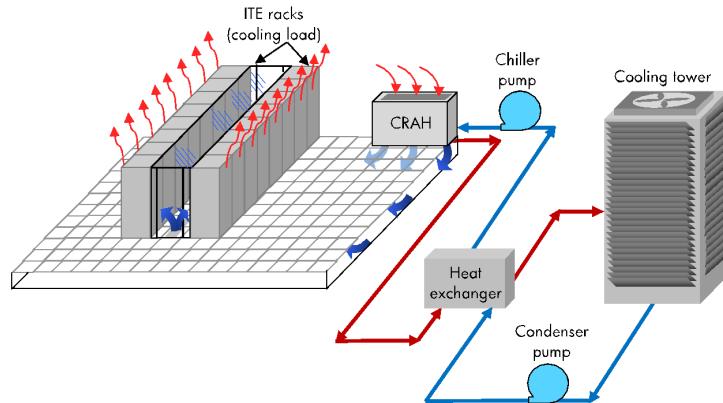


Figura 12: Sistema de refrigeración CRAH

El enfriamiento directo en el chip es la solución más reciente de refrigeración por agua. Esta opción es capaz de refrigerar racks de alta densidad, pero tiene un coste inicial muy alto. Su mayor ventaja es que no necesita enfriar el agua tanto como el resto de opciones, al estar en contacto casi directo con los componentes, lo que reduce los costes. No obstante, los riesgos por fugas o fallos en el sistema aumentan exponencialmente, y si no se utiliza agua destilada puede provocar corrosión en los equipos.

Un ejemplo destacable de la utilización de estos métodos son los centros de datos de *Google*. Los cuales utilizan técnicas que mezclan el aislamiento de pasillos calientes, *free-cooling* y *water-cooling*. En algunos de sus *Data Center* utilizan agua de mar para refrigerar el aire que introducen en los servidores. *Google* calcula la media de los PUE de todos sus centros de datos en 1,11 [10].

Por otra parte, *Facebook* ha conseguido optimizar la refrigeración hasta conseguir PUEs de 1,07 [11] utilizando técnicas diversas, como elevar la temperatura general de sus centros de datos evitando así el *over-cooling*, aislando pasillos fríos y aplicando *water-cooling* y *free-cooling* [20].

Sin embargo, estos datos de *Google* y *Facebook* están fuertemente determinados por las localizaciones que eligen dichas empresas para sus centros de datos, ya que suelen ubicarse en lugares con climas muy fríos. En definitiva, las distintas modalidades de refrigeración por *water-cooling* tiene multitud de inconvenientes para su despliegue en un *Edge Data Center*:

- Requieren de la instalación de nuevas canalizaciones y dispositivos que tienen un consumo energético considerable.
- Requieren de una importante inversión inicial.
- Existe riesgo de fugas o fallos que podrían inutilizar equipos electrónico.
- Son sistemas propensos a provocar corrosión en las piezas que lo forman.
- La densidad de potencia que se puede alcanzar es considerablemente inferior a la que se prevé se podrá alcanzar con la tecnología propuesta en este proyecto.
- Se necesita agua con unas características determinadas y se ha de filtrar para evitar grandes concentraciones de partículas disueltas.
- Necesitan un mantenimiento habitual y un sistema de monitorización.
- Muchas de las opciones de despliegue son modulares, lo cual es válido únicamente para pequeños *Data Centers* y racks de baja densidad.

2.1.3 Refrigeración por inmersión

Tras estudiar las opciones que existen actualmente de refrigeración por aire y por agua, parece razonable pronosticar que ninguna de ellas será adecuada ni eficiente para su instalación en los *Edge Data Center*. Estas soluciones: 1) no logran una alta densidad de potencia por rack, lo que hace que su tamaño aumente demasiado, y 2) la única forma de conseguir un consumo eficiente de climatización es ubicando las instalaciones en lugares con climas fríos, lo que no siempre es posible ya que es necesario emplazarlos cerca de las fuentes de datos.

Ahora vamos a estudiar las opciones que existen hoy en día de refrigeración por inmersión. Los líquidos utilizados deben ser dieléctricos y térmicamente conductores, ya que estos van a estar en contacto directo con los componentes electrónicos. Este tipo de refrigeración tiene el potencial de convertirse en un futuro en la solución más eficaz para

centros de datos, ya que permite reducir drásticamente la energía utilizada en climatización y por lo tanto, reducir el PUE. Las distintas técnicas de refrigeración por inmersión se dividen principalmente en dos tipos:

- Soluciones con líquido refrigerante en **una sola fase**: En ellas el fluido está en constante circulación y arrastra el calor generado por los servidores. El problema es que se necesitan dispositivos externos que enfrien el refrigerante antes de introducirlo nuevamente en el tanque de refrigeración.

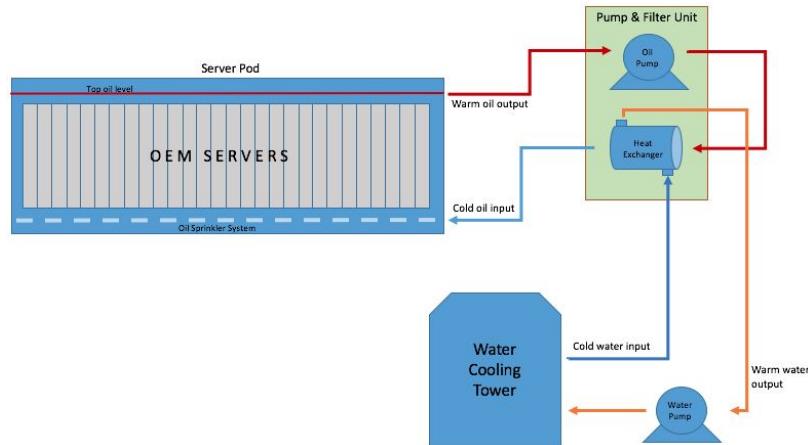


Figura 13: Sistema de refrigeración por inmersión de una fase

- Soluciones con líquido refrigerante en **dos fases**: En estas, el tanque se cierra herméticamente y se intenta aprovechar el calor que absorbe cualquier material, en este caso el refrigerante, al pasar de estado líquido a gaseoso. Cuando esto ocurre, las burbujas de gas que se forman suben a la superficie y se utiliza un condensador o la propia tapa del tanque, para condensar el gas y que pase de nuevo a estado líquido. El principal problema de esta solución es que puede producir presiones muy altas en el sistema, por lo que necesitaremos tanques que soporten estas condiciones.

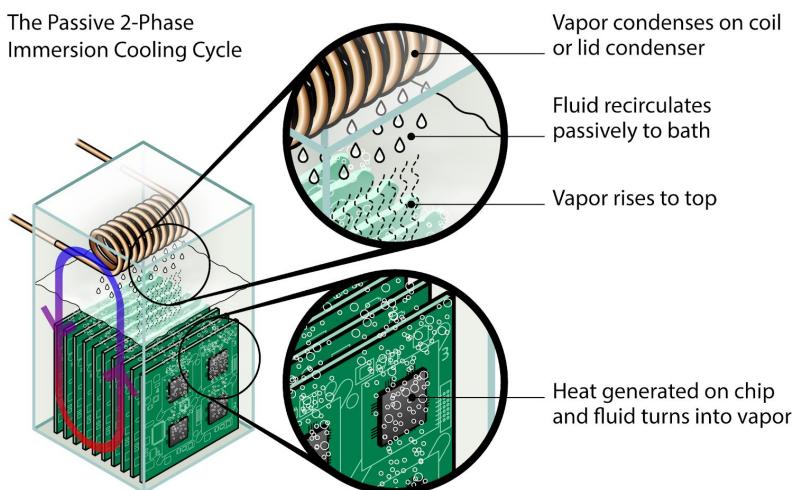


Figura 14: Sistema de refrigeración por inmersión pasiva de dos fases

Existen pocos ejemplos en el mundo de sistemas que utilicen refrigeración por inmersión y la mayoría se utilizan para investigación científica o aplicaciones muy específicas.

El sistema *CarnotJet* de *Green Revolution Cooling* utiliza circulación de aceite mineral a través de un contenedor que contiene las placas base de los servidores sumergidos. Ha sido utilizado desde 2014 en el superordenador *Tsubame KFC*, del *Tokyo Institute of Technology* y fue calificado en dos ocasiones como superordenador más eficiente del mundo según *Green500* [21].



Figura 15: Sistema de refrigeración *Carnotjet* en el supercomputador *Tsubame KFC*

El principal inconveniente de esta solución es lo incómodo y poco higiénico que resulta trabajar en los tanques de aceite. Además funciona en una única fase, por lo que la capacidad de absorción de calor es mucho menor.

Iceotope es una *startup* con sede en Sheffield, Inglaterra, y ofrece soluciones de refrigeración líquida de una fase a nivel de rack. Su sistema se basa en la circulación de un líquido dieléctrico por carcasa herméticas que cubren las placas base de los servidores.



Figura 16: Sistema refrigeración por inmersión de la empresa *Iceotope*

El problema de esta solución es que las carcasa herméticas a nivel de servidor ocupan demasiado espacio, lo que limita la densidad de computación por rack. Además utilizan una única fase, por lo que su capacidad de refrigeración es limitada.

Fujitsu, empresa líder en el mercado de la computación de altas prestaciones, está desarrollando una solución de refrigeración por inmersión en líquido dieléctrico en una fase que ha sido instalada en el *K-Supercomputer* en Japón.

El problema de esta propuesta es que el líquido dieléctrico que utilizan tiene un impacto ambiental no despreciable y que puede generar gases de efecto invernadero. Además, al utilizar una sola fase del líquido, su capacidad de arrastre de calor está bastante limitada.

En cuanto a sistemas de refrigeración en dos fases. existen dos empresas que utilicen un sistema de refrigeración similar a lo que se propone en este proyecto:

1. *BitFury* es una empresa dedicada específicamente a la minería de *Bitcoins*. Ofrece *BlockBox IC*, una solución completa dentro de un contenedor de barco que utiliza un líquido dieléctrico HFE como refrigerante de dos fases.



Figura 17: Sistema *BlockBox IC* de minado de *Bitcoins*

2. EXTOLL es una empresa alemana que fabrica equipos para redes de alto rendimiento, usadas en computación de altas prestaciones. Ofrece *GreenICE*, una solución en un pequeño tanque de inmersión en dos fases de un líquido dieléctrico HFE.



Figura 18: Sistema *GreenICE* de redes de datos de alto rendimiento

Sin embargo, estos productos están orientados a aplicaciones muy específicas, difíciles de adaptar a otros sectores y no optimizadas al máximo para las posibilidades que ofrece el líquido. En este proyecto se plantea analizar más a fondo el sistema para poder dar soluciones más flexibles y optimizadas al máximo para distintos campos de aplicación.

Por todo lo visto hasta ahora en este apartado, podemos concluir que el sistema de refrigeración por inmersión en líquido dieléctrico bi-fase, es una novedad objetiva dentro del

sector de la refrigeración de *Data Centers* que conllevará un importante salto tecnológico y se posiciona como la opción que más ventajas ofrece para los *Edge Data Centers*.

Las aportaciones de este proyecto en concreto contribuirán de manera significativa al desarrollo de sistemas altamente optimizados en climatización para distintas aplicaciones, ya que se pretende aprovechar al máximo las características físicas de arrastre de calor que ofrece el fluido dieléctrico *Novec 7100*.

2.2 Modelos predictivos de temperatura

Ahora vamos a estudiar las distintas técnicas y aproximaciones para la realización de modelos predictivos de temperatura y así obtener conclusiones sobre cuál es la forma más adecuada de realizar el modelo para nuestro sistema.

Como se ha explicado en el apartado de introducción, realizar un modelo predictivo es clave para cualquier sistema electrónico que utilice técnicas de refrigeración y desee trabajar alrededor del punto de máxima eficiencia. Sin embargo, realizar un modelo para sistemas complejos que sea preciso y rápido, puede llegar a ser un intrincado desafío.

2.2.1 Modelos analíticos

Si utilizamos **modelos analíticos**, seremos capaces de representar una solución de forma cerrada, pero requieren la clasificación de todos los parámetros que intervienen e influyen en el sistema, lo cual puede ser una tarea muy tediosa en sistemas complejos. También debemos encontrar las complejas relaciones no lineales entre todos estos parámetros para poder construir una función analítica que recoja todos los factores influyentes.

Estos modelos imponen el uso de características que pueden tener un impacto muy bajo en el objetivo de modelado, degradando así el rendimiento del ajuste de curva. En general, los enfoques analíticos en sistemas complejos reales no consiguen obtener soluciones precisas por la gran cantidad de posibles correlaciones entre factores relevantes, y requieren un análisis y un desarrollo matemático muy complejo que implica una gran cantidad de tiempo y trabajo.

Por ello, en el diseño de muchos modelos se utilizan procedimientos computacionales de alto nivel que puedan ayudar a simplificar la complejidad de los modelos analíticos, como los algoritmos metaheurísticos.

2.2.2 Modelos metaheurísticos

Los **algoritmos metaheurísticos** consisten en la exploración eficiente e iterativa del espacio de búsqueda para conseguir optimizar una función objetivo. En nuestro caso dicha función, podría ser la media del error cometido entre la temperatura real y la predicha.

Estos algoritmos, al contrario que los analíticos, son métodos aproximados. Son especialmente útiles cuando no hay un método exacto de resolución, cuando este requiere

mucho tiempo de cálculo y memoria, o cuando no se necesita la solución óptima y basta con una de buena calidad. Se fundamentan en el uso combinado de conceptos de inteligencia artificial, evolución biológica y mecanismos estadísticos. Esto les da gran flexibilidad y capacidad para resolver una amplia gama de problemas, pero necesitan de grandes cantidades de datos para converger.

Se ha demostrado en numerosas ocasiones prácticas, la gran capacidad de estos algoritmos para resolver satisfactoriamente problemas de alta complejidad computacional con bajo error y en un tiempo razonablemente corto. Entre los algoritmos metaheurísticos más exitosos se encuentran el recocido simulado (*simulated annealing*), la búsqueda tabú (*tabu search*), los algoritmos genéticos (*genetic algorithms*) y las redes neuronales artificiales (*Artificial Neural Networks* o ANN).

Pero tienen un importante inconveniente, y es que a consecuencia de ser algoritmos aproximados, nunca podremos estar completamente seguros de que una solución hallada sea la óptima o si en cambio es una solución subóptima, es decir, un mínimo local.

Además deben alcanzar un compromiso, que en muchas ocasiones es difícil de conseguir, entre una búsqueda diversificada e intensiva de el espacio de búsqueda. Si la búsqueda es demasiado intensiva corre el riesgo de alcanzar únicamente mínimos locales y que la ejecución tarde demasiado tiempo. En cambio si la búsqueda es más diversificada, encontrará rápidamente regiones prometedoras, pero le será muy complicado alcanzar las zonas de mínimo error.

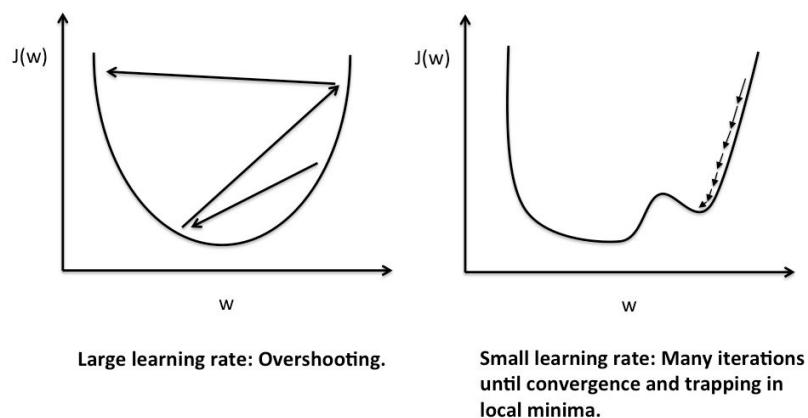


Figura 19: Representación del problema de compromiso entre búsqueda diversificada e intensiva

A pesar de estos posibles problemas, parece razonable concluir que son la mejor opción para realizar un modelo de temperatura de nuestro sistema. Además, estos algoritmos requieren esfuerzo y tiempo considerablemente inferiores que las técnicas analíticas.

2.2.3 Aprendizaje automático y profundo

Por todo lo visto hasta ahora, en este proyecto hemos decidido utilizar algoritmos basados en capas de redes neuronales artificiales (*Artificial Neural Networks* o ANN), como el aprendizaje automático (*Machine Learning*) y aprendizaje profundo (*Deep Learning*). Su

nombre se debe a que tratan de emular el comportamiento de las redes neuronales de los sistemas nerviosos de animales. Estos algoritmos son capaces de aprender de manera automática patrones complejos, correlaciones y comportamientos, analizando grandes cantidades de información. La diferencia entre *Machine Learning* y *Deep Learning* es simplemente la profundidad de la estructura neuronal, es decir, el número de capas de neuronas artificiales.

Las ANN forman un sistema de enlaces entre las distintas neuronas, lo que hace que colaboren entre sí para producir estímulos de salida. Cada enlace posee un peso numérico o *weight*, que se adapta a medida que se entrena la red, y cada neurona contiene una función de activación no lineal. De esta manera, las redes neuronales se adaptan a los impulsos de entrada y son capaces de aprender.

Gracias a las funciones de activación no lineales del interior de las neuronas, las redes son capaces de describir cualquier comportamiento del mundo real, que precisamente es no lineal. Algunos ejemplos típicos son la función sigmoidal, la exponencial normalizada (*SoftMax*), la tangente hiperbólica (*tanh*) o la lineal rectificada (*ReLU*) entre otros.

Dentro de estos algoritmos existen varias modalidades en función de cómo se entrena para que evolucione correctamente:

- *Supervised Learning*: Cuando tenemos un conjunto de datos que incluye los valores objetivo que deseamos predecir. El algoritmo trata de aprender una función que prediga correctamente los valores objetivo a partir del resto de características.
- *Unsupervised Learning*: Cuando tenemos un conjunto de datos pero no hay un objetivo a predecir. El algoritmo debe intentar crear un modelo que podría haber generado dichos datos.
- *Reinforcement Learning*: Cuando tenemos un entorno donde se deben tomar una serie de decisiones secuenciales. Tomar una decisión en cierto instante, influye en las decisiones que se podrán tomar en el futuro. El algoritmo evoluciona gracias a una función de recompensas que le premia cuando está tomando buenas decisiones.

En nuestro caso concreto debemos utilizar *Supervised Learning*, para que el modelo trate de encontrar una función que prediga la temperatura en un futuro, a partir de los datos recogidos en el pasado de diferentes métricas.

En los modelos de *Supervised Learning*, las ANN aprenden utilizando un método matemático llamado retropropagación o *backpropagation*. Es un método de cálculo del gradiente necesario para calcular los pesos de cada enlace en la siguiente iteración del entrenamiento. Su utilización nos permite optimizar la red neuronal para obtener un menor error. Su nombre se debe a que este error se calcula a la salida, comparándola con la solución real (etiqueta o *label*), y de ahí se propaga hacia atrás en la red.

La implementación más básica de las ANN son las redes neuronales simples o ***Simple Neural Network*** (SNN). Estas redes generalizan el concepto de perceptrón simple y su

arquitectura se basa únicamente en la interconexión hacia adelante de neuronas, es decir, las neuronas de una capa se conectan con las neuronas de la siguiente capa (ver Figura 7). De ahí que también reciban el nombre de redes alimentadas hacia delante o redes *feedforward*.

Las redes neuronales convolucionales o ***Convolutional Neural Network*** (CNN) son un tipo de red neuronal donde las células corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico. Son muy efectivas para tareas de visión artificial como clasificación y segmentación de imágenes entre otras.

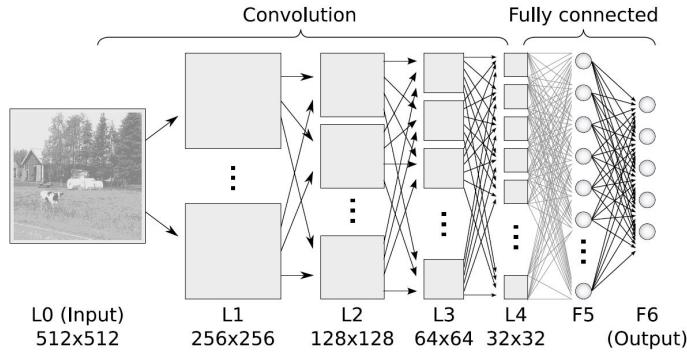


Figura 20: Esquema de una red neuronal convolucional

Otra implementación más compleja, y la que nos interesa para nuestro caso concreto, son las redes neuronales recurrentes o ***Recurrent Neural Networks*** (RNN). En este tipo de redes las conexiones forman grafos dirigidos a lo largo de una secuencia, lo que les permite exhibir un comportamiento temporal dinámico. Además algunas neuronas que la forman son más complejas, pudiendo hacer uso de un estado interno o memoria. Existen varios tipos de neuronas dentro de las RNN:

- *Simple RNN cell:*

Estas neuronas son similares a las de las redes *feedforward*, pero se les añade un factor de retroalimentación variable en el tiempo. El problema de este tipo de neuronas es que en la fase de *backpropagation* del error, el gradiente puede llegar a ser multiplicado un gran número de veces (tantos como instantes temporales) lo que puede provocar que los pesos de los enlaces obtengan valores demasiado pequeños o demasiado grandes para un aprendizaje satisfactorio. Esto provoca una gran inestabilidad y esto motivó el desarrollo de las neuronas LSTM y GRU.

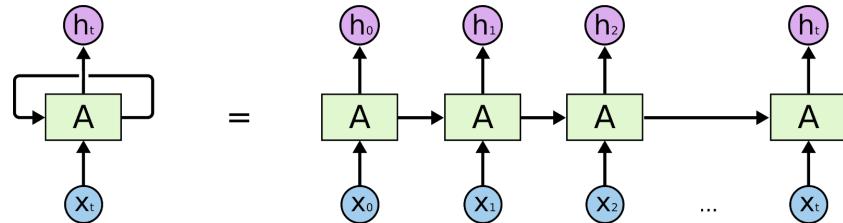


Figura 21: Esquema de una neurona *SimpleRNN* desenrollada en la secuencia temporal

- LSTM:

En este tipo de neuronas se añade una entrada de “olvido” a la célula, que le permite recordar el estado en el que se encontraba en instantes de tiempo anteriores. Solucionando así los problemas asociados a la *Simple RNN cell*.

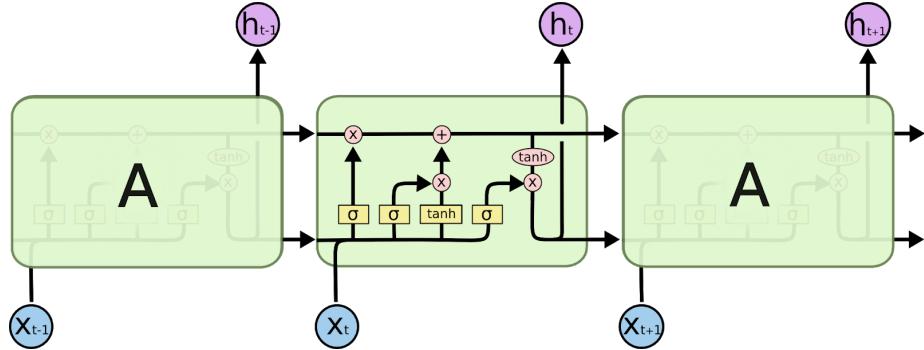


Figura 22: Esquema de una neurona LSTM desenrollada en la secuencia temporal

- GRU:

Esta neurona surgió más tarde que las LSTM (2014) y trata de minimizar sus parámetros para disminuir el tiempo de entrenamiento. Se ha demostrado que tienen un mejor rendimiento para *datasets* de menor tamaño.

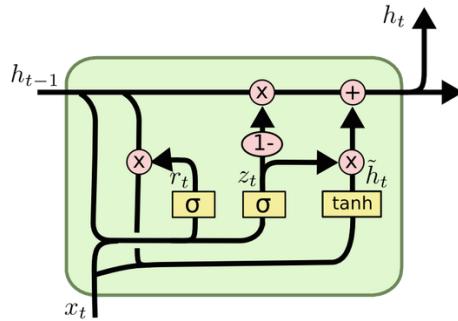


Figura 23: Esquema de una neurona GRU

Realizando una búsqueda de artículos de investigación científica no se han encontrado publicaciones sobre modelos de temperatura de un sistema distribuido refrigerado por inmersión. Sin embargo podemos extrapolar algunos métodos y estructuras que han utilizado en otros trabajos de investigación acerca de predicciones temporales.

Según diversas investigaciones de instituciones científicas de renombre como el *IEEE* [22], las redes neuronales recurrentes (RNN) responden de manera muy satisfactoria a problemas de predicción temporal. Esto es porque en las RNN las conexiones entre nodos forman un grafo dirigido a lo largo de una secuencia, lo que permite que aprendan también de comportamientos temporales dinámicos. Y a diferencia del resto de redes neuronales, algunas pueden utilizar un estado interno o memoria para procesar las secuencias de entrada.

Investigadores de la *Faculty of Science and Technology* de Marruecos, desarrollaron un modelo de predicción meteorológica utilizando redes neuronales recurrentes de tipo LSTM [23]. En el que conseguían predecir la temperatura, humedad y velocidad del viento con errores alrededor del 2%.

En la revista científica *Nature* se publicó una investigación en la que se mostraba un ejemplo de predicción temporal multivariable con valores faltantes utilizando redes neuronales recurrentes de tipo GRU [24]. Esta aplicación es especialmente interesante cuando se trabaja con señales biológicas, que suelen ser ruidosas e inestables. Se consiguió predecir los valores faltantes con un error medio alrededor de 0,7 y una desviación estándar de 0,02.

Investigadores del *IEEE* desarrollaron un modelo que predecía la demanda de taxis en Nueva York [25] con redes basadas en neuronas LSTM. En este trabajo se conseguían precisiones de acierto cercanas al 85%.

El *Central Research Institute of Electric Power Industry* de Japón publicó una investigación en la que se pretendía predecir la demanda de potencia eléctrica de nueve instalaciones industriales utilizando RNN [26]. Se conseguían predecir con errores de entre 0,53% y 2,76%.

Y existen numerosos ejemplos más acerca de RNN utilizadas satisfactoriamente en predicción temporal. Por lo tanto, parece razonable tomar la decisión de utilizar este tipo de estructuras neuronales en nuestro proyecto de predicción de temperatura.

3. Solución

En este apartado se presenta la solución propuesta al problema planteado en el apartado de introducción.

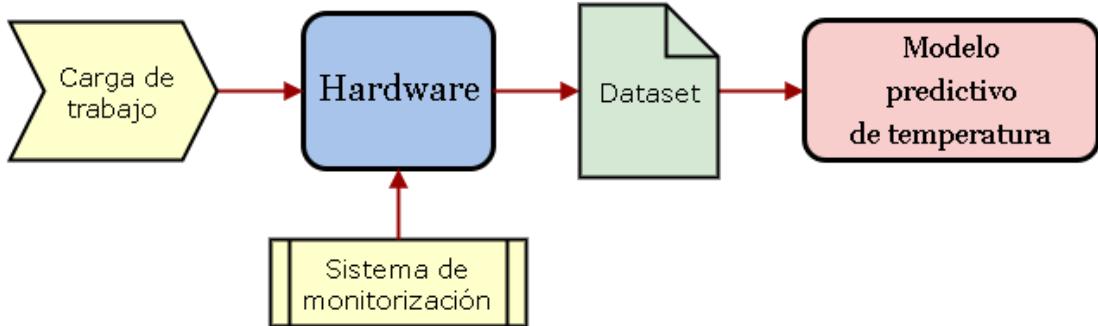


Figura 24: Esquema de la solución

3.1 Montaje físico del sistema

El prototipo se basa en placas *Raspberry Pi 3 Model B+* que son las encargadas de ejecutar la carga de trabajo, capturar datos en tiempo real y exportarlos a la plataforma de almacenamiento y visualización. Su elección se debe principalmente a su bajo coste, pequeño tamaño y facilidad de uso.

Se hace uso de tres placas colocadas en forma de cluster utilizando espaciadores de 1,5 cm para poder medir la influencia entre los equipos electrónicos cercanos en el modelo de predicción de temperatura. Cada placa necesita una tarjeta de memoria SD en la que se ha instalado el sistema operativo *Raspbian* [27].

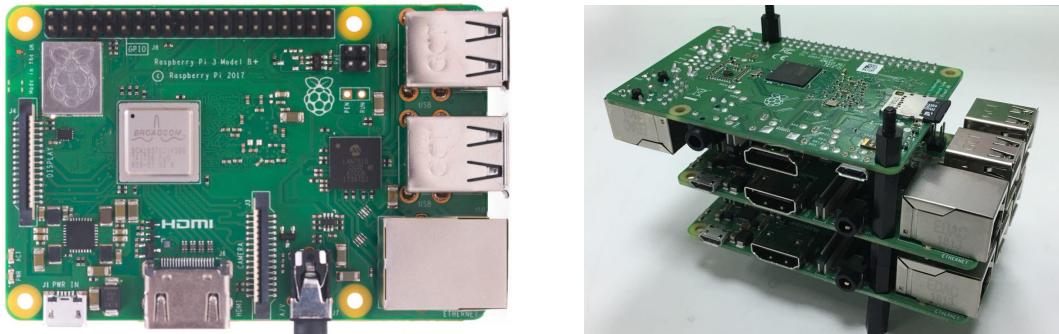


Figura 25: Placa *Raspberry Pi 3 Model B+* y cluster de 3 placas

El cluster se sumerge en el líquido dieléctrico *Novec 7100*, utilizando una pecera de cristal. Todas las placas necesitan conexión a Internet para realizar la transmisión de datos, por lo que se debe instalar un *switch Ethernet* junto a la pecera.

Para la conexión a la toma de corriente eléctrica se necesitan alimentadores con salida de 5V y 2,5A, según las especificaciones de la *Raspberry Pi 3 Model B+* [28]. Se colocarán

todas las placas en posición vertical, para que las burbujas de gas generadas puedan subir sin obstáculos a la superficie.

Se ha diseñado el prototipo para que las placas al calentarse, evaporen el líquido dieléctrico, este se condense en la tapa y se precipite de nuevo en estado líquido. Aunque debemos tener en cuenta las limitaciones técnicas de la *Raspberry Pi*, es una placa de prestaciones de gama media por lo que le será difícil llegar al punto térmico de máxima eficiencia. En el futuro, si se demuestra la viabilidad de esta solución, se realizarán experimentos con placas de mayores prestaciones y mayor potencia.

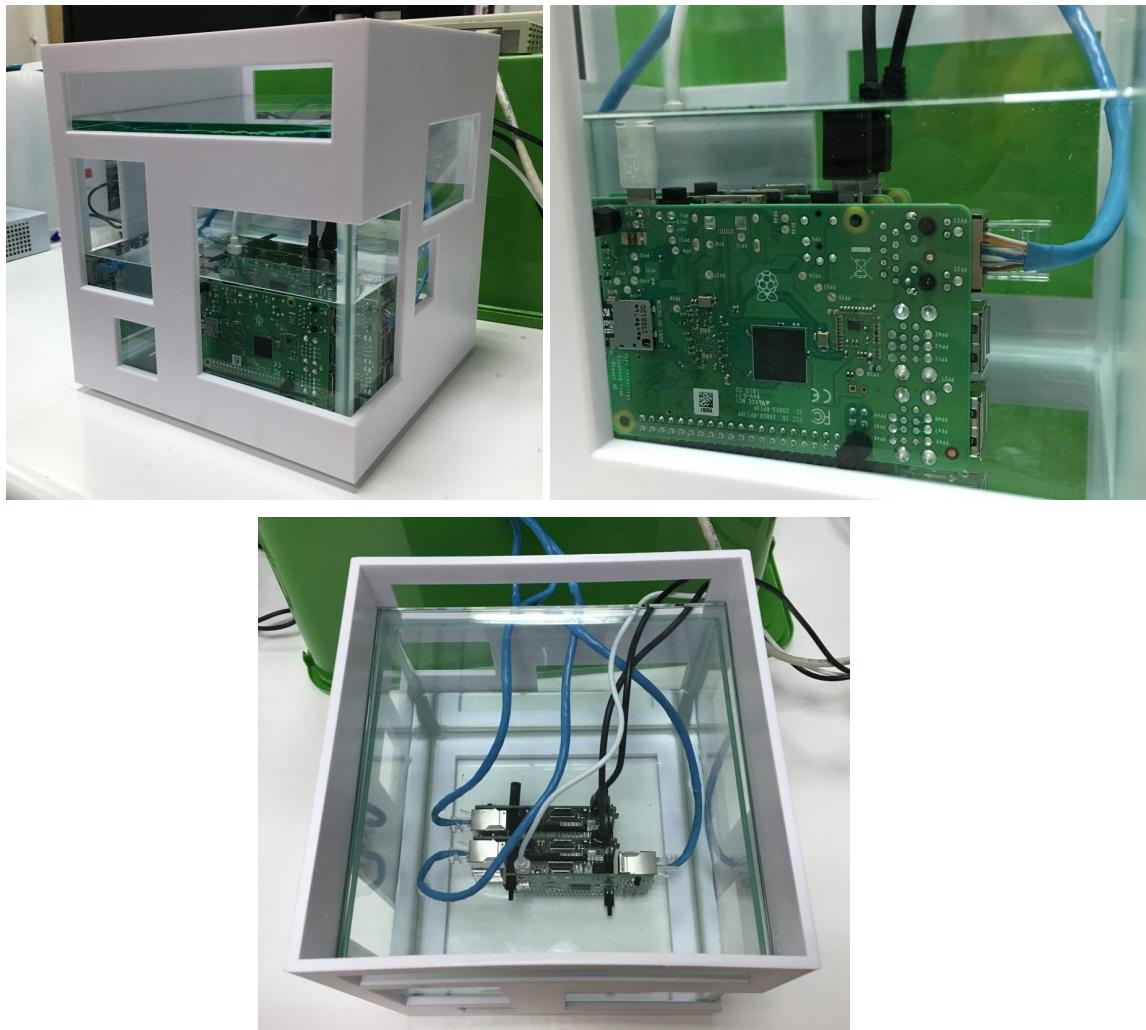


Figura 26: Fotografías del prototipo

3.2 Diseño de la carga de trabajo

Para simular un entorno real de *Edge Data Center* en una *Smart City*, se ha decidido que la carga de trabajo será la ejecución de *Data Analytics* con datos reales de una ciudad. En concreto las placas *Raspberry Pi* deberán ejecutar el entrenamiento de un modelo predictivo del nivel de contaminación del aire en la ciudad de Pekín, China. El *dataset* lo proporciona de manera gratuita la *University of California, Irvine* [29]. Este incluye la evolución temporal durante 5 años de los siguientes datos y medidas:

- Fecha y hora.
- PM2,5: Concentración de partículas en suspensión de menos de 2,5 micras (medida estándar de contaminación ambiental).
- DEWP: Punto de rocío, que depende de la temperatura y la humedad relativa.
- Temperatura.
- Presión atmosférica.
- CBWD: Dirección combinada del viento.
- LSW: Velocidad del viento acumulada.
- LS: Horas de nevada acumuladas.
- LR: Horas de lluvia acumuladas.

Para la ejecución se utiliza una estructura neuronal de células LSTM basada en el trabajo de investigación realizado también en la *University of California* [30]. Cabe destacar que lo que se pretende es generar distintos patrones de cargas reales que presenten perfiles reales de temperaturas en ejecución. Es decir, que no evaluaremos la calidad u optimización del trabajo realizado por las placas *Raspberry Pi*.

A la hora de programar la carga de trabajo, se ha realizado un banco de pruebas que se repite cada 4 horas y 45 minutos. Se considera buena práctica hacer que todos los parámetros varíen en algún momento de la ejecución, para así obtener un *dataset* robusto y que el modelo aprenda a predecir correctamente evitando el sobreajuste u *overfitting*.

Para observar la influencia de la frecuencia de trabajo de CPU, la ejecución se divide en varios periodos, en los que se ejecuta la carga imponiendo a la CPU una frecuencia de trabajo de 600 MHz o de 1,4 GHz. Estas son las dos únicas frecuencias a las que puede trabajar el procesador de la *Raspberry Pi 3 Model B+*. Para variar dichas frecuencias de los cores de la CPU utilizamos el paquete *cpufrequtils* [31], incluído en el sistema operativo *Raspbian* [27], mediante los siguientes comandos en el terminal:

```
>> sudo cpufreq-set -r -g userspace  
>> sudo cpufreq-set -r -f 600MHz  
ó  
>> sudo cpufreq-set -r -f 1.4GHz
```

El modo “*userspace*” nos permite seleccionar entre las frecuencias permitidas de los cores y al incluir la opción “-r” realizamos el cambio en todos ellos simultáneamente.

Cada uno de los períodos se divide en 4 ejecuciones cíclicas cuya duración está entre 30 y 40 minutos. Por lo tanto para observar la dependencia de la temperatura objetivo, la que queremos predecir, con las temperaturas de las placas *Raspberry Pi* a los lados en el cluster, alternamos las ejecuciones. Se ha decidido utilizar una placa a cada lado debido a que no son simétricas y, como pretendemos aumentar la densidad lo máximo posible, la configuración del cluster tampoco será simétrica. Entonces la influencia de cada placa será distinta. La secuencia temporal de ejecuciones es la siguiente:

Nº ejec. cíclica y duración	Frecuencia CPU	<i>Raspberry Pi</i> Nº 1	<i>Raspberry Pi</i> Nº 2	<i>Raspberry Pi</i> Nº 3
1 (35 min)	600 MHz	<i>Sleep</i>	<i>Execute</i>	<i>Sleep</i>
2 (30 min)		<i>Execute</i>	<i>Execute</i>	<i>Sleep</i>
3 (40 min)		<i>Execute</i>	<i>Execute</i>	<i>Execute</i>
4 (35 min)		<i>Sleep</i>	<i>Execute</i>	<i>Execute</i>
5 (40 min)	1,4 GHz	<i>Execute</i>	<i>Execute</i>	<i>Sleep</i>
6 (30 min)		<i>Sleep</i>	<i>Execute</i>	<i>Execute</i>
7 (35 min)		<i>Sleep</i>	<i>Execute</i>	<i>Sleep</i>
8 (40 min)		<i>Execute</i>	<i>Execute</i>	<i>Execute</i>

En cada ejecución cíclica se pretende observar la dependencia de la temperatura objetivo, con el porcentaje de utilización de CPU. Por tanto, cada una se divide en 5 períodos en los que variaremos el número de hilos de ejecución o *threads*. El porcentaje de utilización de CPU evoluciona temporalmente según el siguiente ejemplo:



Figura 27: Porcentaje de utilización de CPU en cada ejecución cíclica

- 1: Ejecutando 1 *thread* (~ 65% CPU utilizada)
- 2: Ejecutando 2 *threads* (~ 80% CPU utilizada)
- 3: Ejecutando 4 *threads* (~ 90% CPU utilizada)
- 4: Ejecutando 2 *threads* (~ 80% CPU utilizada)
- 5: Ejecutando 1 *thread* (~ 65% CPU utilizada)

3.3 Software de captura y visualización

Para la captura de datos se ha decidido utilizar la herramienta *Collectd*. Este software, que es un proyecto *free open-source*, nos permite recolectar y transferir periódicamente las métricas del sistema que deseemos. Las transmisiones son de tipo UDP, es decir, sin confirmación. El sistema enviará los datos sin preocuparse de si han llegado correctamente al destino. Una vez obtenidos los datos y deseemos entrenar el modelo, se realizará un preprocessado de estos para evitar posibles valores nulos. Este software será instalado en las placas *Raspberry Pi* en las que utilizamos el sistema operativo *Raspbian* [27].

El modelo predictivo de temperatura debe predecir la temperatura en instantes futuros de la placa intermedia del cluster. Por lo tanto, después de hacer un pequeño estudio acerca de las métricas del sistema disponibles para transmitir y que utilizaremos más tarde para realizar el modelo, se han elegido las siguientes:

- Sensor de temperatura de la placa lateral izquierda del cluster. [°C]
- Porcentaje de utilización de CPU de la placa intermedia del cluster. [0% - 100%]
- Frecuencia de trabajo de CPU de la placa intermedia del cluster. [600 MHz, 1,4 GHz]
- Sensor de temperatura de la placa intermedia del cluster. [°C]
- Sensor de temperatura de la placa lateral derecha cluster. [°C]

Para el almacenamiento y visualización de los datos se ha decidido utilizar *Graphite*. Esta herramienta *free open-source*, monitoriza y representa gráficamente datos numéricos en el tiempo. Su elección se debe principalmente a su facilidad de uso y a que este software ya estaba instalado en el servidor web de nuestro grupo de investigación, *GreenLSI*. [31]

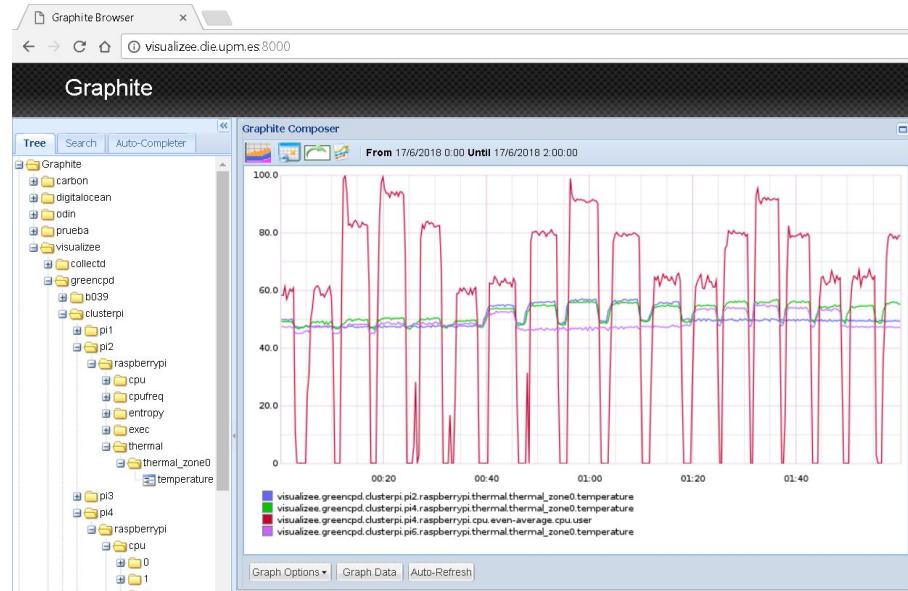


Figura 28: Servidor web de *GreenLSI* con la herramienta *Graphite*

3.4 Modelo predictivo de temperatura

3.4.1 Herramientas de software utilizadas

A partir del *dataset* obtenido con las cargas de trabajo en el sistema físico montado, implementamos el modelo predictivo de temperatura utilizando distintas herramientas software. Para la creación de las estructuras neuronales utilizamos *Keras* [11], que es una librería de redes neuronales de alto nivel escrita en lenguaje *Python*. Fue desarrollada para permitir una implementación rápida, sencilla y modular de estructuras neuronales. Además desde 2017 *Google* soporta y contribuye a esta librería.

Para su ejecución es necesario que trabaje sobre librerías de computación neuronal de más bajo nivel como *TensorFlow*, *CNTK* o *Theano*. En nuestro caso hemos escogido *TensorFlow* [12], que es una librería de código abierto creada por *Google*, debido a la gran cantidad de documentación, facilidad de uso, por estar muy optimizada y a que soporta *Python*.

También hacemos uso de librerías de *Python* para el manejo de matrices N-dimensionales como *numpy* [33] y *pandas* [34], la librería *scikit-learn* [35] que nos da acceso a una gran variedad de operaciones matemáticas de análisis masivo de datos y el paquete *h5py* [35] para exportar en archivos los modelos creados.

3.4.2 Hiperparámetros

En el desarrollo e implementación de los modelos predictivos con técnicas de *Machine Learning* y *Deep Learning* aparecen un gran número de parámetros de alto nivel que, como diseñadores del modelo, podemos modificar hasta encontrar los valores óptimos para nuestro caso concreto. Los más importantes son:

- **Nº features:** Cantidad de características distintas que introducimos al modelo. En nuestro caso vamos a utilizar 5 (Temperatura placa 1, Temperatura placa 2, Porcentaje de utilización de CPU placa 2, Frecuencia CPU placa 2, Temperatura placa 3)
- **Batch size:** Número de instantes temporales introducidos a la red neuronal antes de que se reajusten los pesos mediante *backpropagation*. Si su valor es demasiado alto, la red no tendrá tiempo suficiente para aprender. Y si es demasiado bajo no aprenderá satisfactoriamente la evolución temporal.
- **Epochs:** Número de veces que se introduce el *dataset* a la red neuronal para que realice el entrenamiento. Si este parámetro es demasiado pequeño no tendrá tiempo suficiente de aprender (*underfitting*) y si es demasiado grande la red puede memorizar los resultados del *dataset* concreto y no generalizar la predicción para otros casos (*overfitting*).
- **Ventana de predicción:** Intervalo temporal que predice el modelo. A mayor ventana de predicción peor calidad tendrá esta, pero menos útil resultará el modelo. Debemos encontrar un compromiso entre este parámetro y la calidad de la predicción.
- **Porcentaje de test:** Reparto de la cantidad de datos que utiliza el modelo para el entrenamiento y para el test.
- **Preprocesado de datos:** Tratamiento previo de los datos antes de introducirlo a la red neuronal. (Escalado, normalización, etc.)
- **Estructura de la red:** Número, tamaño y tipo de capas neuronales en la red.
- **Función de activación:** Función de respuesta no lineal de las neuronas. (*tanh*, *ReLU*, *softmax*, etc.)
- **Función de cálculo del error:** Función mediante la que se calcula el error a la salida de la red neuronal. (MSE, MAE, MSLE, *CrossEntropy*, etc.)
- **Algoritmo de optimización:** Algoritmo con el que se calcula los nuevos pesos de los enlaces en función del error, en la *backpropagation*. (SGD, RMSprop, Adagrad, Adam, etc.)

3.4.3 Métricas de evaluación de modelos predictivos

Existen diversos indicadores estadísticos para verificar el rendimiento y calidad del modelo predictivo. Los más utilizados en los ejemplos del estado del arte, y que por tanto hemos decidido escoger, son los siguientes:

- **Error medio absoluto ± Desviación estándar (MAE ± STD °C):**

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - x_j| \quad STD = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \mu)^2}$$

- **Porcentaje de error medio absoluto (MAPE %):**

$$MAPE = \frac{1}{n} \sum_{j=1}^n \left| \frac{y_j - x_j}{y_j} \right| \cdot 100\%$$

- **Coeficiente de determinación (R² %)**

$$R^2 = \frac{\sum_{j=1}^n (y_j - x_j)^2}{\sum_{j=1}^n (y_j - \bar{y}_j)^2}$$

- **Error máximo absoluto (MáxEA °C):**

$$MáxAE = máx \{ |y_j - x_j|, \forall j \in [1, n] \}$$

- **Error máximo relativo (MáxAPE °C):**

$$MáxAPE = máx \left\{ \left(\frac{|y_j - x_j| \cdot 100\%}{y_j} \right), \forall j \in [1, n] \right\}$$

- **Desviación de la media cuadrática (RMSD °C)**

$$RMSD = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - x_j)^2}$$

4. Experimentos

En este apartado se describen todas las pruebas realizadas las con distintas configuraciones y estructuras del modelo predictivo. Las fases de experimentación y optimización serán las siguientes:

1. A partir de ejemplos del estado del arte, definir unos valores para los hiperparámetros y decidir cuáles serán fijos y cuáles variables. Además, debemos comprobar que esta configuración converge a una solución aceptable que será nuestro modelo base.
2. Experimentar con los distintos optimizadores y funciones de error para seleccionar los más óptimos.
3. Realizar pruebas variando la estructura y tamaño de la red, para encontrar la que mejor resultados ofrece.
4. Tras esto experimentar con el resto de hiperparámetros variables hasta encontrar la configuración con mejores resultados de predicción.
5. Finalmente realizar pruebas con distintas ventanas temporales y encontrar un compromiso entre tiempo de predicción y error cometido.

El orden de las fases descritas se justifica en que la estructura de la red neuronal es el factor más primario porque define el modelo. Una vez definido un modelo con buenos resultados, variamos otros hiperparámetros que intervengan en el entrenamiento y añadimos capas *Dropout* para optimizarlo. Calculamos previamente el optimizador y la función de error debido a que tienen un alto grado de independencia con el resto de hiperparámetros. Y finalmente, con la configuración y estructura óptimas, realizamos pruebas con distintas ventanas temporales para comprobar su funcionamiento.

Realizaremos los pasos 3, 4 y 5 del procedimiento para los tres tipos de estructuras recurrentes vistas anteriormente (*SimpleRNN*, LSTM y GRU).

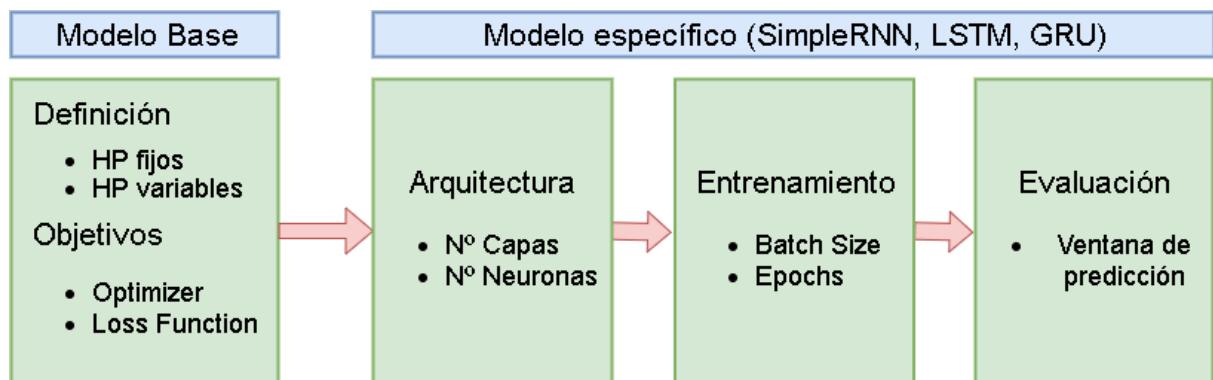


Figura 29: Esquema de los experimentos

4.1 Modelo base

Comenzamos por tanto eligiendo los valores de los hiperparámetros basándonos en ejemplos del estado del arte y los ofrecidos por la librería *keras* [37]:

- **Hiperparámetros fijos**
 - **Nº features:** 5 (Temperatura placa 1, Temperatura placa 2, Porcentaje de utilización de CPU placa 2, Frecuencia CPU placa 2, Temperatura placa 3)
 - **Porcentaje de test:** 12% (Se recomienda un porcentaje bajo para *datasets* pequeños)
 - **Preprocesado de datos:** Escalado $\in [0, 1]$ (Se considera una buena práctica normalizar los valores introducidos en la red neuronal)
- **Hiperparámetros variables**
 - **Función de cálculo del error:** MAE
 - **Algoritmo de optimización:** Nadam
 - **Función de activación:** *tanh*
 - **Batch size:** 65
 - **Epochs:** 80
 - **Ventana de predicción:** 1 minuto (Se considera un tiempo suficiente para tomar medidas directas o indirectas sobre la temperatura del *Edge Data Center*)
 - **Estructura de la red:** 2 capas LSTM de 128 y 64 neuronas respectivamente

Tras entrenar el modelo observamos que ofrece resultados razonablemente positivos:

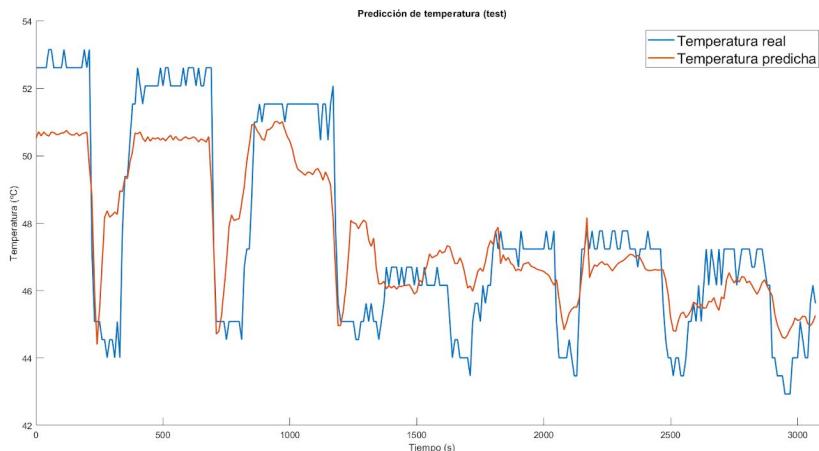


Figura 30: Predicción de temperatura del modelo inicial basado en LSTM

MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
0.96 \pm 1.093	2.058	71.883	6.362	14.457	1.455

A continuación vamos a experimentar con todos los optimizadores [38] y funciones de cálculo de error [39] que implementa la librería *keras*.

- **Función de cálculo del error**

<i>Loss Function</i>	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
MSE	1.866 \pm 1.288	3.919	42.787	6.165	14.009	2.267
MAE	1.313 \pm 0.929	2.758	71.186	4.954	11.258	1.609
MAPE	4.696 \pm 2.997	9.501	- 245.3	10.168	19.13	5.570
MSLE	2.061 \pm 1.437	4.263	29.741	6.718	15.265	2.513
Log cosh	1.848 \pm 1.24	3.83	44.904	5.877	13.355	2.225
Binary CrossEntropy	1.676 \pm 1.205	3.538	52.579	5.982	13.593	2.064

La función de cálculo del error que mejores resultados ofrece es el **error medio absoluto (MAE)**. Por tanto la utilizaremos en el resto de experimentos.

- **Algoritmos de optimización**

Optimizador	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
SGD	1.465 \pm 1.445	3.186	52.844	5.962	13.223	2.058
RMSprop	1.308 \pm 1.036	2.784	69.011	5.368	12.198	1.668
Adagrad	1.848 \pm 1.587	4.021	33.93	6.281	14.111	2.437
Adadelta	2.449 \pm 1.436	5.262	10.305	6.815	15.677	2.839
Adam	1.405 \pm 1.034	2.977	66.124	5.119	11.541	1.745
Adamax	1.23 \pm 1.211	2.642	66.825	5.288	11.871	1.727
Nadam	1.313 \pm 0.929	2.758	71.186	4.954	11.258	1.609

El optimizador con mejores resultados es **Nadam**. Por tanto la utilizaremos en el resto de experimentos.

4.2 Red basada en *SimpleRNN*

4.2.1 Estructura neuronal

Ahora vamos a experimentar distintas estructuras para las redes basadas en SimpleRNN. Según los ejemplos del estado del arte, se considera buena práctica que el número de neuronas en las capas sean potencias de dos (2, 4, 8, 16, etc.).

Estructura (nº neuronas)	MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
1 capa (16)	1.039 ± 1.063	2.222	75.405	6.796	15.443	1.486
1 capa (64)	1.04 ± 1.13	2.244	73.741	6.598	14.992	1.536
1 capa (128)	1.107 ± 0.925	2.374	76.835	5.56	12.635	1.443
1 capa (256)	1.09 ± 0.832	2.33	79.071	4.941	11.227	1.371
2 capas (8, 8)	1.089 ± 1.213	2.354	70.397	5.906	13.422	1.631
2 capas (16, 8)	1.131 ± 1.078	2.416	72.824	6.385	14.51	1.562
2 capas (64, 16)	1.118 ± 1.125	2.404	71.978	6.582	14.956	1.586
2 capas (32, 32)	1.089 ± 1.251	2.357	69.38	6.904	15.688	1.659
2 capas (128, 32)	1.259 ± 1.265	2.705	64.546	7.631	17.341	1.785
3 capas (32, 16, 8)	1.08 ± 1.181	2.32	71.47	7.116	16.17	1.601
3 capas (16, 32, 16)	1.106 ± 1.139	2.377	71.925	7.346	16.693	1.588
3 capas (32, 32, 16)	1.188 ± 1.145	2.552	69.667	6.874	15.621	1.651
3 capas (64, 32, 16)	1.155 ± 1.162	2.486	70.11	6.716	15.262	1.639

La estructura neuronal que mejores resultados ofrece es **1 capa con 256 neuronas**. A continuación, utilizando esta estructura, vamos a intentar optimizar el resto de hiperparámetros.

4.2.2 Función de activación

Activation	MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
<i>Softmax</i>	1.127 ± 0.895	2.42	76.934	4.375	9.943	1.439
<i>Elu</i>	1.133 ± 1.115	2.442	71.878	6.304	14.325	1.589
<i>Selu</i>	1.636 ± 1.357	3.497	49.725	7.287	16.56	2.125

<i>Softplus</i>	1.606 ± 1.048	3.481	59.059	4.56	10.362	1.918
<i>Softsign</i>	1.083 ± 1.102	2.314	73.42	6.68	15.179	1.545
<i>Relu</i>	1.022 ± 1.252	2.197	70.923	8.238	16.185	1.616
<i>tanh</i>	1.09 ± 0.832	2.33	79.071	4.941	11.227	1.371
<i>Sigmoid</i>	1.635 ± 1.125	3.562	56.155	4.863	11.052	1.985
<i>Hard Sigmoid</i>	1.395 ± 0.963	3.013	68.025	4.833	10.982	1.695
<i>Linear</i>	1.192 ± 0.841	2.535	76.3	5.038	11.448	1.46

La función de activación con mejores resultados es la tangente hiperbólica (**tanh**).

4.2.3 *Batch Size*

<i>Batch Size</i>	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
15	1.303 ± 0.788	2.798	65.478	3.784	8.496	1.523
25	0.958 ± 0.767	2.063	76.74	3.77	8.464	1.227
35	0.897 ± 0.739	1.93	79.535	3.696	8.297	1.162
45	1.043 ± 0.92	2.251	74.366	5.12	11.635	1.39
55	1.1 ± 0.73	2.34	76.7	3.807	8.547	1.32
65	1.09 ± 0.832	2.33	79.071	4.941	11.227	1.371
75	1.08 ± 0.81	2.314	74.62	3.941	8.847	1.35
85	1.289 ± 1.031	2.714	72.17	5.44	12.362	1.651
95	0.885 ± 0.919	1.918	81.402	3.794	8.362	1.276
105	0.931 ± 0.881	2.013	76.77	3.893	8.739	1.282
115	1.141 ± 1.044	2.455	70.454	5.008	11.38	1.546

El *Batch size* con mejores resultados es **95**.

4.2.4 *Epochs*

<i>Epochs</i>	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
20	0.92 ± 0.93	1.989	80.464	3.829	8.596	1.308
30	0.933 ± 0.952	2.022	79.705	4.089	8.658	1.333

40	0.933 ± 0.963	2.022	79.47	3.941	8.681	1.341
50	0.908 ± 0.944	1.966	80.402	4.012	8.562	1.31
60	0.901 ± 0.928	1.952	80.88	4.007	8.484	1.294
70	0.901 ± 0.931	1.952	80.819	3.964	8.419	1.296
80	0.885 ± 0.919	1.918	81.402	3.794	8.362	1.276
90	0.881 ± 0.902	1.906	81.85	3.773	8.47	1.261
100	0.911 ± 0.933	1.975	80.577	3.868	8.685	1.304
110	0.897 ± 0.896	1.94	81.632	3.719	8.35	1.268
120	0.885 ± 0.895	1.916	81.896	3.743	8.403	1.259
130	0.855 ± 0.89	1.851	82.59	3.754	8.429	1.235
140	0.836 ± 0.842	1.8	83.897	3.688	8.281	1.187
150	0.889 ± 0.902	1.924	81.663	3.872	8.694	1.267
160	0.854 ± 0.89	1.848	82.619	3.764	8.449	1.234
170	0.884 ± 0.882	1.905	82.188	3.919	8.799	1.249
180	0.919 ± 0.859	1.974	81.908	3.875	8.699	1.259

El número de *Epochs* con mejores resultados es **140**. Por tanto la red basada en *SimpleRNN* optimizando los hiperparámetros (*Epochs* = 140, *Batch size* = 95, función de activación = *tanh*) realiza la siguiente predicción:

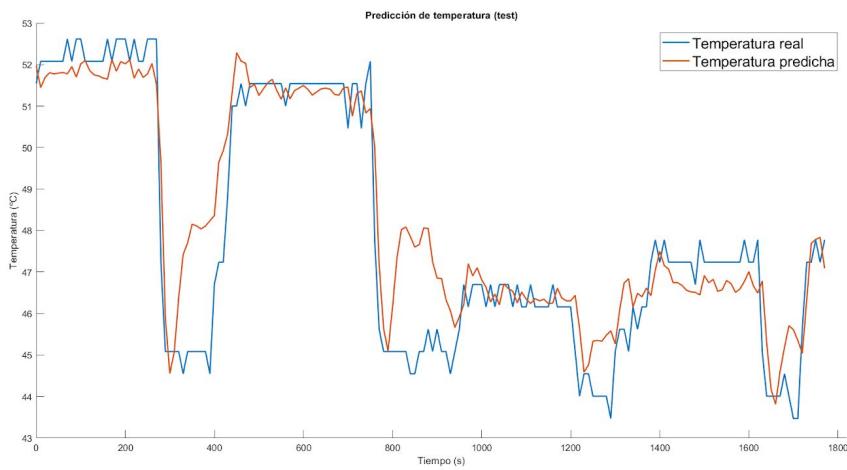


Figura 31: Predicción de red basada en *SimpleRNN* tras optimizar hiperparámetros

MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
0.836 ± 0.842	1.8	83.897	3.688	8.281	1.187

4.3 Red basada en LSTM

4.3.1 Estructura neuronal

Ahora vamos a experimentar distintas estructuras para las neuronas tipo LSTM.

Estructura (nº neuronas)	MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
1 capa (1)	1.2 ± 0.968	2.57	73.553	4.793	10.891	1.542
1 capa (2)	0.911 ± 0.886	1.959	82.044	4.444	9.875	1.27
1 capa (4)	1.008 ± 1.111	2.182	74.976	5.589	12.701	1.5
1 capa (8)	0.855 ± 1.116	1.853	78.005	6.579	14.949	1.406
1 capa (16)	0.973 ± 1.136	2.089	75.108	7.123	16.185	1.496
1 capa (32)	0.967 ± 1.17	2.085	74.351	7.316	16.623	1.518
1 capa (64)	0.96 ± 1.158	2.07	74.82	6.713	15.255	1.504
2 capas (1, 1)	1.022 ± 0.822	2.198	80.871	4.426	9.803	1.311
2 capas (4, 2)	0.935 ± 1.161	2.018	75.263	6.381	14.5	1.491
2 capas (8, 4)	0.978 ± 1.275	2.124	71.257	6.276	14.261	1.607
2 capas (16, 8)	0.947 ± 1.18	2.04	74.547	6.876	15.625	1.512
2 capas (32, 16)	0.973 ± 1.196	2.094	73.559	7.09	16.11	1.542
3 capas (4, 2, 1)	0.98 ± 1.195	2.113	73.43	6.491	14.75	1.545
3 capas (8, 4, 2)	0.948 ± 1.131	2.035	75.782	6.418	14.583	1.475
3 capas (16, 8, 4)	0.986 ± 1.159	2.116	74.232	7.046	16.011	1.522

La estructura neuronal que mejores resultados ofrece es **1 capa con 2 neuronas**. A continuación procedemos a optimizar el resto de hiperparámetros.

4.3.2 Función de activación

Activation	MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
<i>Softmax</i>	1.295 ± 1.033	2.766	69.461	4.952	11.252	1.657
<i>Elu</i>	0.969 ± 0.949	2.098	79.54	4.405	10.011	1.356
<i>Selu</i>	1.135 ± 0.979	2.449	75.007	5.103	11.596	1.499
<i>Softplus</i>	1.33 ± 0.952	2.817	70.242	5.025	11.418	1.635

<i>Softsign</i>	1.302 ± 1.012	2.769	69.763	4.98	11.316	1.649
<i>Relu</i>	1.079 ± 0.852	2.306	78.965	4.209	9.441	1.375
<i>tanh</i>	0.911 ± 0.886	1.959	82.044	4.444	9.875	1.27
<i>Sigmoid</i>	1.289 ± 0.904	2.737	72.426	4.939	11.224	1.574
<i>Hard Sigmoid</i>	1.12 ± 0.945	2.402	76.127	4.693	10.665	1.465
<i>Linear</i>	0.99 ± 1.185	2.152	73.462	6.284	14.28	1.544

La función de activación con mejores resultados es la tangente hiperbólica (**tanh**).

4.2.3 *Batch Size*

<i>Batch Size</i>	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
5	0.879 ± 0.928	1.907	72.13	5.27	11.83	1.278
15	0.976 ± 1.134	2.12	66.699	5.667	12.723	1.496
25	0.86 ± 0.774	1.856	79.321	3.909	8.53	1.157
35	0.892 ± 0.776	1.922	78.811	3.819	8.086	1.183
45	1.003 ± 1.085	2.18	71.078	4.614	10.486	1.478
55	1.174 ± 0.755	2.51	73.928	3.839	8.516	1.396
65	0.911 ± 0.886	1.959	82.044	4.444	9.875	1.27
75	0.964 ± 0.786	2.073	78.466	3.429	7.606	1.244
85	1.127 ± 1.431	2.445	66.128	5.635	12.805	1.822
95	0.952 ± 1.001	2.041	78.232	4.312	9.128	1.381
105	0.883 ± 0.824	1.902	79.4	3.794	8.518	1.208
115	1.083 ± 0.958	2.334	74.194	4.131	8.936	1.445

El *Batch size* con mejores resultados es **65**.

4.3.4 *Epochs*

<i>Epochs</i>	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
20	1.087 ± 0.869	2.356	74.267	3.983	8.835	1.392
30	1.043 ± 0.895	2.219	74.881	4.912	11.162	1.375

40	1.108 ± 0.891	2.369	73.135	4.76	10.816	1.422
50	1.158 ± 0.822	2.469	73.201	4.491	10.204	1.42
60	1.167 ± 0.839	2.489	72.551	4.653	10.572	1.437
70	1.057 ± 0.861	2.258	75.286	4.691	10.66	1.364
80	0.911 ± 0.886	1.959	82.044	4.444	9.875	1.27
90	1.113 ± 0.834	2.373	74.307	4.649	10.563	1.39
100	1.003 ± 1.001	2.136	73.298	5.601	12.728	1.417
110	0.769 ± 0.798	1.663	83.661	4.12	8.723	1.109
120	0.764 ± 0.685	1.646	85.999	4.334	9.175	1.026
130	0.727 ± 0.692	1.568	86.606	4.637	9.817	1.004
140	1.022 ± 0.875	2.183	79.862	4.331	9.842	1.345
150	1.053 ± 0.911	2.259	78.44	4.543	10.322	1.392
160	0.88 ± 1.067	1.912	78.702	5.091	10.778	10.778
170	0.834 ± 0.99	1.808	77.726	5.472	12.433	1.295
180	0.714 ± 0.725	1.543	86.239	4.858	10.285	1.018
190	0.799 ± 0.897	1.717	80.826	5.086	11.557	1.201
200	0.735 ± 0.675	1.584	86.779	4.326	9.159	0.997
210	0.711 ± 0.658	1.535	87.527	4.535	9.6	0.969
220	0.688 ± 0.767	1.492	85.904	5.238	11.088	1.03
230	0.812 ± 1.081	1.756	75.717	5.858	13.312	1.352
240	0.865 ± 1.123	1.874	73.283	5.268	11.971	1.418
250	0.841 ± 1.128	1.827	73.676	5.155	11.713	1.407
260	0.848 ± 1.142	1.837	73.106	5.878	13.357	1.423

El número de *Epochs* con mejores resultados es **210**. Por tanto la red basada en LSTM optimizando los hiperparámetros (*Epochs* = 210, *Batch size* = 65, función de activación = *tanh*) realiza la siguiente predicción:

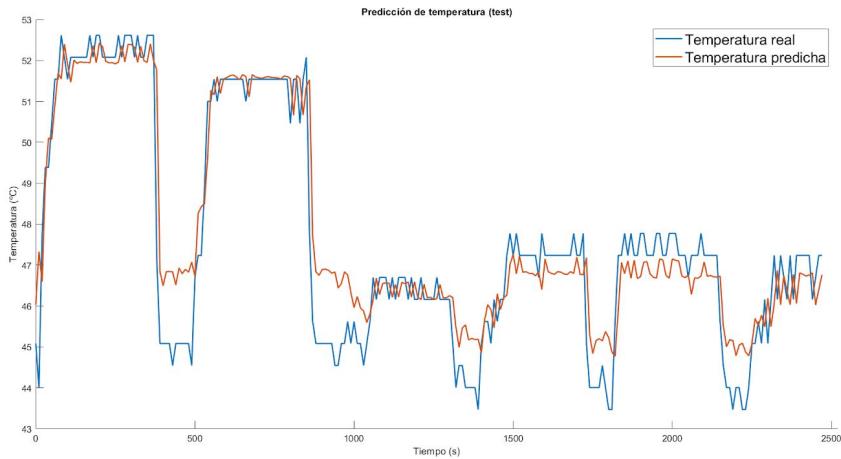


Figura 32: Predicción de red basada en LSTM tras optimizar hiperparámetros

MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
0.711 ± 0.658	1.535	87.527	4.535	9.6	0.969

4.4 Red basada en GRU

4.4.1 Estructura neuronal

Ahora vamos a experimentar distintas estructuras para las neuronas tipo GRU.

Estructura (nº neuronas)	MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
1 capa (2)	0.891 ± 0.717	1.902	82.621	3.949	8.974	1.144
1 capa (4)	0.74 ± 0.675	1.59	86.668	4.483	10.187	1.002
1 capa (8)	0.793 ± 0.996	1.719	78.473	5.906	13.421	1.273
1 capa (16)	0.79 ± 1.016	1.71	77.981	6.417	14.582	1.287
1 capa (32)	0.907 ± 1.072	1.951	73.8	7.126	16.192	1.404
1 capa (64)	0.918 ± 1.146	1.975	71.361	7.117	16.173	1.468
2 capas (2, 1)	0.918 ± 0.76	1.963	81.129	4.15	9.429	1.192
2 capas (4, 2)	0.847 ± 0.778	1.821	82.429	4.687	10.651	1.15
2 capas (8, 4)	0.883 ± 1.127	1.913	72.76	6.65	15.112	1.432
2 capas (16, 8)	0.941 ± 1.046	2.025	73.684	6.73	15.292	1.407
2 capas (32, 16)	1.069 ± 1.088	2.281	69.079	6.945	15.781	1.525
3 capas (2, 1, 1)	0.963 ± 0.728	2.05	80.647	4.589	10.428	1.207

3 capas (2, 2, 1)	0.842 ± 0.839	1.818	81.231	4.207	9.56	1.188
3 capas (4, 2, 1)	0.806 ± 0.83	1.737	82.211	4.656	10.581	1.157
3 capas (8, 4, 2)	0.829 ± 0.979	1.784	78.15	6.246	14.192	1.282
3 capas (16, 8, 4)	0.877 ± 1.076	1.891	74.399	6.651	15.114	1.388
3 capas (32, 16, 8)	0.896 ± 1.113	1.925	72.86	6.705	15.235	1.429
3 capas (8, 16, 8)	0.922 ± 1.081	1.987	73.188	6.964	15.825	1.42
3 capas (4, 8, 2)	0.837 ± 1.064	1.811	75.635	7.064	16.052	1.354

La estructura neuronal que mejores resultados ofrece es **1 capa con 4 neuronas**. A continuación procedemos a optimizar el resto de hiperparámetros.

4.4.2 Función de activación

Activation	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
Softmax	1.063 ± 0.792	2.264	76.649	4.185	9.51	1.326
Elu	0.718 ± 0.808	1.55	84.483	4.688	10.652	1.081
Selu	0.837 ± 0.739	1.805	83.436	4.071	9.251	1.116
Softplus	0.86 ± 0.831	1.867	80.984	4.033	9.164	1.196
Softsign	0.847 ± 0.733	1.817	83.324	4.594	10.44	1.12
Relu	0.912 ± 0.771	1.961	81.037	4.067	9.241	1.195
tanh	0.74 ± 0.675	1.59	86.668	4.483	10.187	1.002
Sigmoid	1.116 ± 0.857	2.377	73.687	4.342	9.867	1.407
Hard Sigmoid	0.995 ± 0.887	2.136	76.385	4.418	10.04	1.333
Linear	0.89 ± 1.076	1.916	74.079	6.164	14.006	1.397

La función de activación con mejores resultados es la tangente hiperbólica (**tanh**).

4.4.3 Batch Size

Batch Size	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
15	1.13 ± 0.896	2.423	69.072	5.593	12.556	1.442
25	0.804 ± 0.867	1.732	78.424	4.439	9.965	1.182

35	0.773 ± 0.781	1.677	81.707	3.882	8.219	1.099
45	1.054 ± 1.263	2.279	64.155	6.455	14.668	1.645
55	0.915 ± 0.78	1.966	80.654	4.166	9.351	1.202
65	0.74 ± 0.675	1.59	86.668	4.483	10.187	1.002
75	0.751 ± 0.684	1.625	85.638	4.156	8.799	1.016
85	0.998 ± 1.187	2.158	75.451	5.472	12.435	1.551
95	0.726 ± 0.798	1.571	86.714	4.512	9.552	1.079
105	1.027 ± 0.865	2.205	74.538	3.89	8.732	1.343
115	1.069 ± 1.133	2.319	70.032	5.118	11.629	1.558

El *Batch size* con mejores resultados es **65**.

4.4.4 Epochs

Epochs	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
10	1.067 ± 0.964	2.293	76.395	3.729	8.27	1.438
20	1.184 ± 0.795	2.497	76.797	3.667	8.231	1.426
30	1.09 ± 0.978	2.333	75.518	4.176	9.376	1.465
40	1.032 ± 0.875	2.196	79.119	3.641	8.174	1.353
50	0.918 ± 0.781	1.952	83.432	3.671	8.24	1.205
60	0.916 ± 0.901	1.957	81.163	4.572	9.679	1.285
70	1.137 ± 0.966	2.429	74.614	4.173	9.367	1.491
80	0.726 ± 0.798	1.571	86.714	4.512	9.552	1.079
90	0.833 ± 0.857	1.796	83.69	3.831	8.11	1.195
100	0.837 ± 0.788	1.791	84.934	3.684	8.27	1.149
110	0.764 ± 0.817	1.652	85.722	4.185	8.861	1.119
120	0.753 ± 0.592	1.611	89.539	2.822	6.195	0.957
130	0.756 ± 0.789	1.636	86.376	4.264	9.027	1.093
140	0.912 ± 1.118	1.965	76.229	4.587	10.297	1.443
150	0.919 ± 1.015	1.973	78.613	4.9	11.0	1.369
160	0.9 ± 1.046	1.933	78.278	4.462	10.017	1.38

El número de *Epochs* con mejores resultados es **120**. Por tanto la red basada en LSTM optimizando los hiperparámetros (*Epochs* = 120, *Batch size* = 95, función de activación = *relu*) realiza la siguiente predicción:

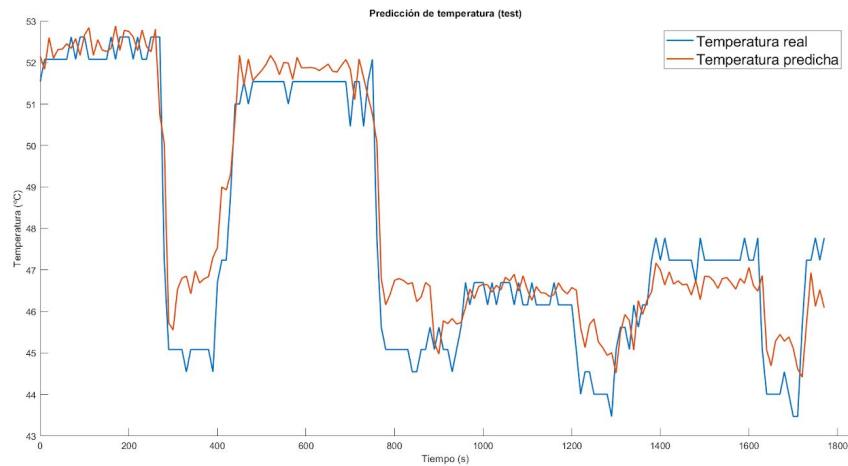


Figura 33: Predicción de red basada en GRU tras optimizar hiperparámetros

MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
0.753 \pm 0.592	1.611	89.539	2.822	6.195	0.957

5. Resultados

5.1 Comparación de modelos

En este apartado se presentan los resultados de los tres tipos de arquitecturas optimizadas en el apartado de experimentación.

Tipo de red	MAE \pm STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
SimpleRNN	0.836 \pm 0.842	1.8	83.897	3.688	8.281	1.187
LSTM	0.711 \pm 0.658	1.535	87.527	4.535	9.6	0.969
GRU	0.753 \pm 0.592	1.611	89.539	2.822	6.195	0.957

Si mostramos en una misma gráfica las tres predicciones, queda de la siguiente manera:

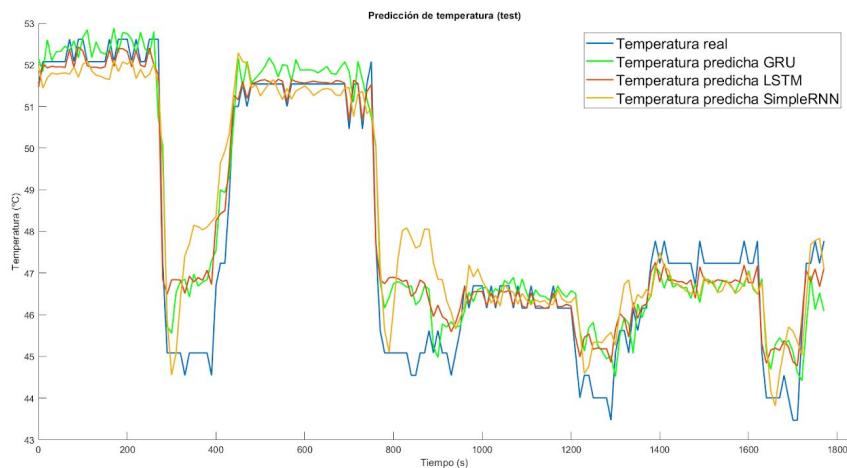


Figura 34: Comparación de predicciones de los tres tipos de redes neuronales

Alrededor del segundo 850 en el test, la frecuencia de la CPU cambia de 1.4GHz a 600MHz. Este hecho va a ser muy difícil de predecir para el modelo, a no ser que sea algo cíclico. Por ello, alrededor de esos instantes el modelo predice subidas de temperatura que no ocurren. Pero cabe destacar que una vez el modelo obtiene como dato el cambio de frecuencias, responde satisfactoriamente a la predicción de nuevas temperaturas. Por lo que podemos deducir que ha aprendido correctamente la dependencia de la temperatura con la frecuencia de trabajo de la CPU.

5.2 Modelo seleccionado

Tras los experimentos podemos concluir que la estructura neuronal con mejores resultados para la predicción de 1 minuto es la basada en GRU.

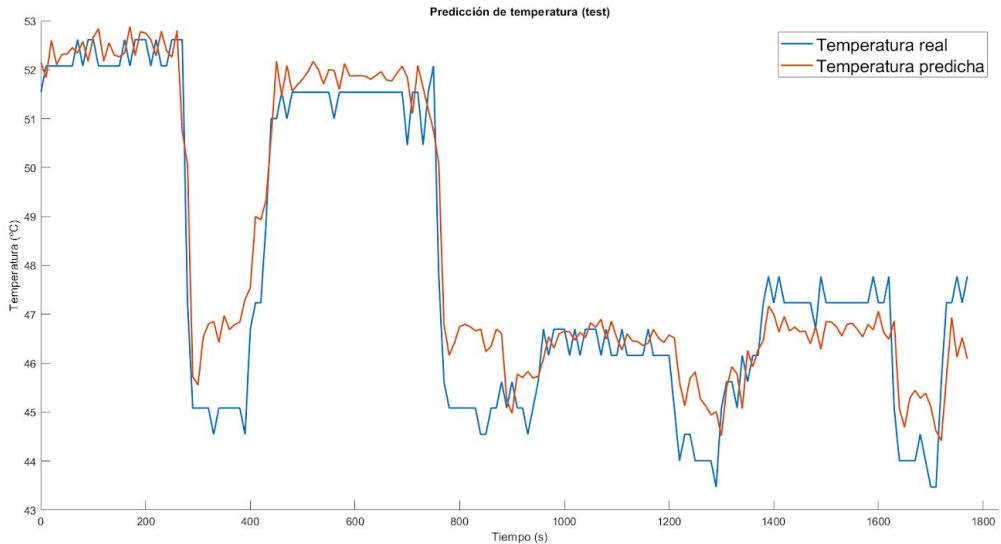


Figura 35: Predicción de la red neuronal basada en GRU con ventana temporal de 1 minuto

MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
0.753 ± 0.592	1.611	89.539	2.822	6.195	0.957

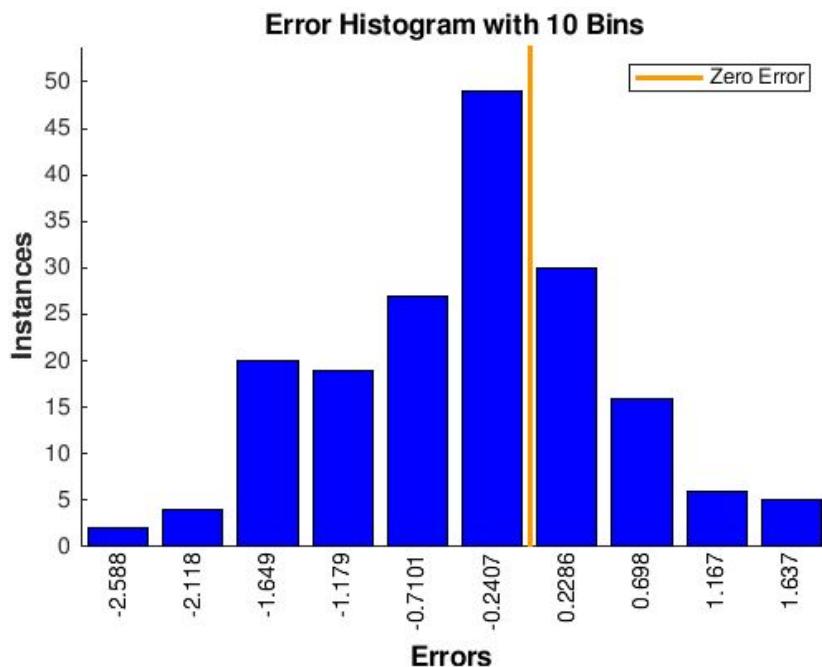


Figura 36: Histograma de errores en la predicción de la red neuronal basada en GRU

A pesar de que se ha optimizado para una ventana temporal de 1 minuto, vamos a experimentar con otros tiempos de predicción y de esta forma poder comprobar la sensibilidad de este parámetro en el modelo.

Ventana temporal	MAE ± STD °C	MAPE %	R ² %	MáxAE °C	MáxAPE %	RMSD °C
10 s	0.473 ± 0.587	1.01	92.504	5.2	11.008	0.754
20 s	0.477 ± 0.564	1.023	92.779	5.036	10.661	0.738
30 s	0.529 ± 0.546	1.133	92.326	5.014	10.614	0.76
40 s	0.547 ± 0.511	1.178	92.558	3.279	7.273	0.748
50 s	0.618 ± 0.564	1.336	90.651	3.142	7.053	0.836
60 s	0.753 ± 0.592	1.611	89.539	2.822	6.195	0.957
70 s	0.863 ± 0.879	1.868	82.339	3.583	7.806	1.232
80 s	1.117 ± 1.003	2.412	73.087	4.292	9.636	1.502
90s	1.159 ± 1.122	2.498	67.944	5.069	11.378	1.613
100 s	1.399 ± 1.295	3.026	53.482	5.713	12.826	1.906
110 s	1.364 ± 1.449	2.961	47.518	5.517	12.385	1.99
120 s	1.563 ± 1.512	3.381	34.536	5.734	12.873	2.175

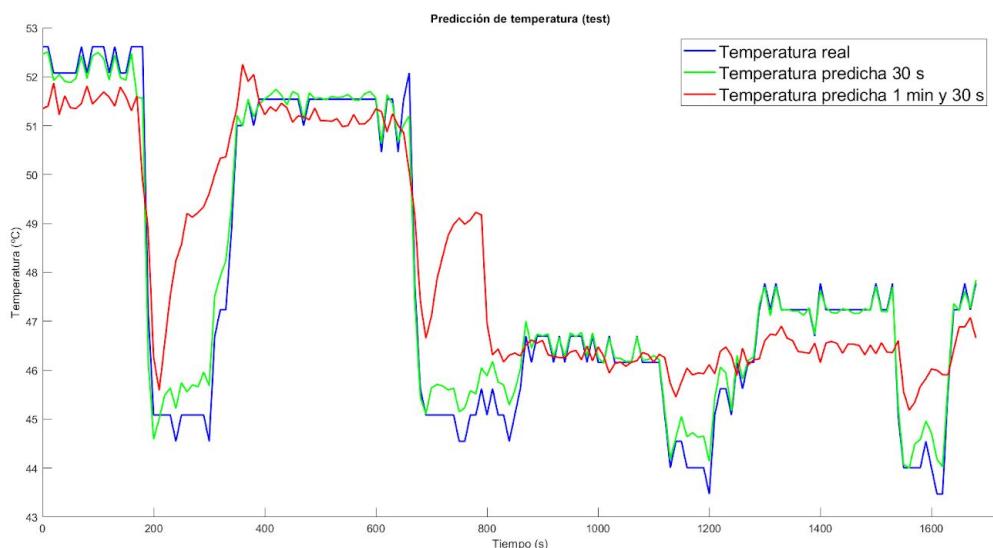


Figura 37: Comparación de predicciones con distintas ventanas temporales

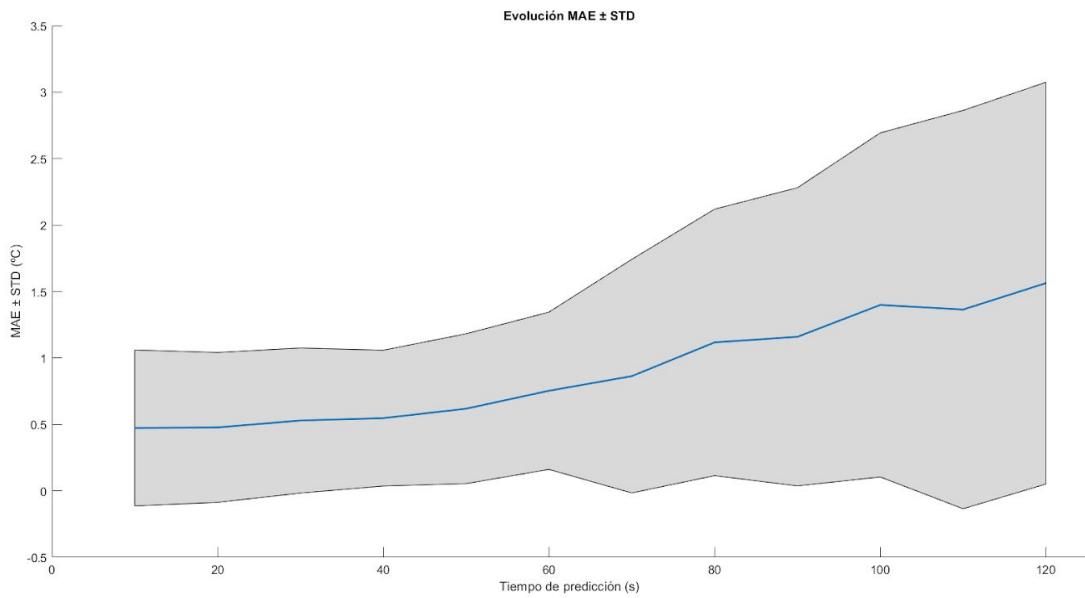


Figura 38: Evolución del error medio absoluto y desviación estándar (MAE ± STD) en función de la ventana de predicción

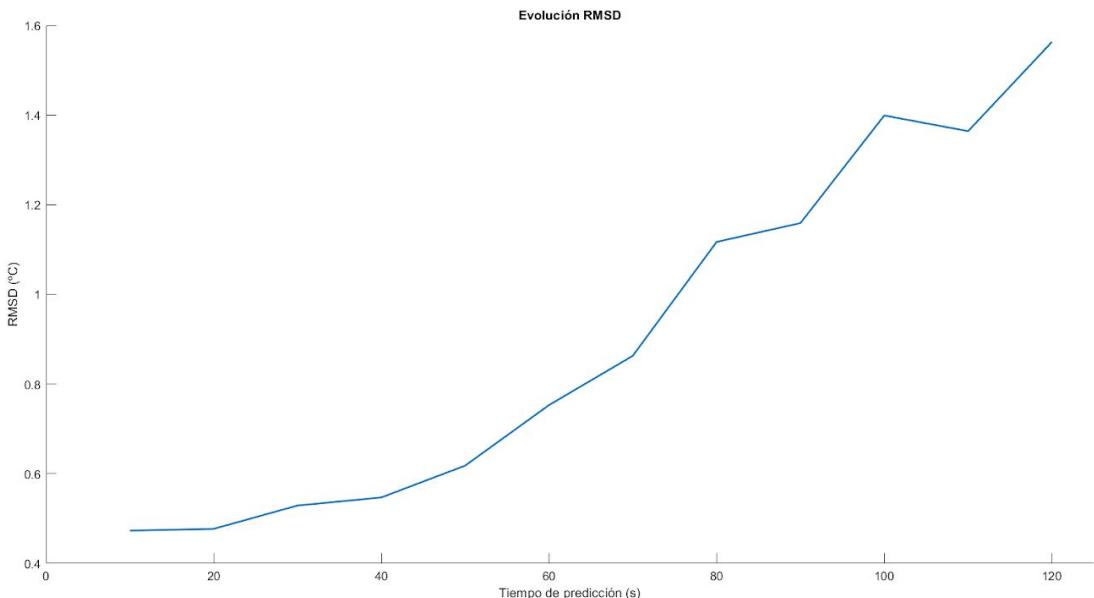


Figura 39: Evolución de la desviación de la media cuadrática (RMSD) en función de la ventana de predicción

Se puede concluir que se ha desarrollado un modelo predictivo con un error entorno a 0.753 °C y una ventana de predicción de 1 minuto, suficientemente alta para encontrar un compromiso específico para sistemas que necesiten una predicción de temperatura a corto o medio plazo. Y gracias a la facilidad del entrenamiento de las redes neuronales artificiales, este método es adaptable para multitud de sistemas con distintas especificaciones, haciendo uso únicamente el histórico de datos capturados.

6. Conclusiones y líneas futuras

En este proyecto se ha diseñado e implementado con éxito un modelo predictivo basado en una arquitectura de red neuronal artificial recurrente, con neuronas tipo GRU (*Gated Recurrent Unit*). Dicho modelo realiza satisfactoriamente predicciones acerca de la temperatura de un sistema basado en refrigeración por inmersión en un fluido Hidro-Fluoro-Éter (HFE). Con una ventana temporal de 1 minuto el modelo predice con media de error de 0.753 °C, una desviación estándar de 0.592 °C y un error máximo de 2.822 °C.

Utilizando este método se pueden realizar predicciones con una antelación suficiente para tomar medidas directas o indirectas sobre la propia temperatura del sistema. Esto permite que el fluido refrigerante se encuentre en un rango específico de temperaturas donde optimiza la capacidad de arrastre de calor.

También se han estudiado las distintas posibilidades existentes en la actualidad para la implantación de *Edge Data Centers* núcleos urbanos. Se ha llegado a la conclusión de que la mejor opción son sistemas basados en refrigeración por inmersión pasiva bifase en líquidos dieléctricos, como el *Novec 7100*. Las principales razones que nos han llevado a dicha conclusión son:

- 1) La alta densidad de potencia que se puede llegar a conseguir (de hasta 250 kW por rack) en comparación con otros métodos de refrigeración, lo que permite reducir en gran medida el espacio ocupado por el centro de datos.
- 2) La eliminación del consumo energético de climatización (que hoy en día alcanza el 40%) por ser una refrigeración pasiva. Esto permite que se puedan alcanzar tasas de PUE (*Power usage effectiveness*) iguales a 1, por lo que las redes eléctricas de las ciudades podrán soportar sin grandes esfuerzos su implantación.
- 3) La alta capacidad de arrastre de calor del fluido refrigerante al realizar el cambio de fase a estado gaseoso.

Cabe destacar que esta solución no se basa en una simple mejora del estado actual de la técnica, sino que cambia el paradigma por completo, destacando sus innovadores aspectos tecnológicos, conceptuales y metodológicos, además de su influencia estratégica potencial en el sector tecnológico y de Data Centers.

Las aportaciones de este proyecto en concreto contribuirán de manera significativa al desarrollo de sistemas altamente optimizados que utilicen este sistema de refrigeración para distintas aplicaciones, ya que se pretende aprovechar al máximo las características físicas de arrastre de calor que ofrece el fluido dieléctrico HFE.

Se han encontrado problemas cuando se ha intentado llegar al punto de temperatura óptimo para arrastre de calor del líquido refrigerante, debido a las bajas prestaciones de las placas *Raspberry Pi 3 Model B+*, que no son capaces de alcanzar altas temperaturas. Por tanto, las líneas futuras de esta solución pasan por la fabricación de un tanque de inmersión sellado donde se realicen experimentos con placas base de servidores profesionales que permitan, a la mayor brevedad posible, el despliegue de *Edge Data Centers* que utilice esta tecnología de refrigeración, en entornos urbanos.

A pesar de este problema, se considera que el modelo predictivo es una solución completamente válida gracias a la facilidad de adaptación y despliegue de las redes neuronales artificiales. Por ello se propone como línea futura la optimización, automatizado y paralelización del entrenamiento de modelos predictivos en tiempo real, que reduzcan al mínimo la intervención humana en el mantenimiento de estos sistemas.

Este proyecto se ha desarrollado utilizando el fluido dieléctrico *Novec 7100*, pero existen multitud de opciones de líquidos refrigerantes con características similares en el mercado. Por tanto, se propone también como línea futura de investigación, experimentación y comparación con otros fluidos o mezclas de varios de ellos, que permita encontrar el más óptimo en arrastre de calor, precio e impacto medio ambiental.

7. Bibliografía

- [1] United Nations, New York, 2015, «World urbanization prospects», [En línea]. Available: <https://esa.un.org/unpd/wup/Publications/Files/WUP2014-Report.pdf>
- [2] Estudio Universidad Carolina del Norte, «Climate change expected to increase premature deaths from air pollution,» [En línea]. Available: <https://uncnews.unc.edu/2017/07/31/climate-change-expected-increase-premature-deaths-air-pollution>
- [3] Agencia Internacional de Energía, «World Energy Outlook 2017,» [En línea]. Available: http://www.iea.org/media/weowebsite/2017/Chap1_WEO2017.pdf
- [4] Agencia Internacional de Energía, «World Energy Outlook 2017 Executive Summary,» [En línea]. Available: https://www.iea.org/publications/freepublications/publication/WEO_2017_Executive_Summary_English_version.pdf
- [5] J. Howell, «Number of Connected IoT Devices Will Surge to 125 Billion by 2030, IHS Markit,» [En línea]. Available: <https://technology.ihs.com/596542/number-of-connected-iot-devices-will-surge-to-125-billion-by-2030-ihs-markit-says>
- [6] A. Shilov, «Intel forms new group for autonomous vehicles and announces \$250M Investment,» [En línea]. Available: <https://www.anandtech.com/show/10872/intel-forms-new-group-for-autonomous-vehicles-and-announces-250m-investment>
- [7] N. Engbers and E. Taen, «Green Data Net. Report to IT Room INFRA,» *European Commision*. FP7 ICT 2013.6.2, Nov. 2014
- [8] Data Center Knowledge, «Special Report: The World's Largest Data Centers,» [En línea]. Available: <http://www.datacenterknowledge.com/special-report-the-worlds-largest-data-centers/>
- [9] T. J. Breen, E. J. Walsh, J. Punch, A. J. Shah and C. E. Bash, «From chip to cooling tower data center modeling: Part I Influence of server inlet temperature and temperature rise across cabinet,» *2010 12th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, Las Vegas, NV, 2010, pp. 1-10. doi: 10.1109/ITHERM.2010.5501421
- [10] Google Data Centers, «Eficiencia: cómo lo hacemos,» [En línea]. Available: <https://www.google.com/about/datacenters/efficiency/internal/#water-and-cooling>
- [11] J. Park, «Designing a Very Efficient Data Center, Facebook Data Centers,» [En línea]. Available: <https://code.facebook.com/posts/1398612007031339/designing-a-very-efficient-data-center/>

- [12] 3M, «Fluido especial 3M™ Novec™ 7100,» [En línea]. Available: https://www.3m.com/es/3M/es_ES/empresa-es/todos-productos-3m/~/Fluido-especial-3M-Novec-7100/?N=5002385+8709318+8709341+8710650+8710710+8711017+8717595+8736412+8745549+3294002049&rt=rud
- [13] 3M, «3M Novec 7100 Fluid Used in World's Largest Two-Phase Immersion Cooling Project,» [En línea]. Available: <http://news.3m.com/press-release/company/3m-novec-7100-fluid-used-worlds-largest-two-phase-immersion-cooling-project>
- [14] Keras, «Keras: The Python Deep Learning library,» [En línea]. Available: <https://keras.io/>
- [15] «Tensorflow, An open source machine learning framework for everyone,» [En línea]. Available: <https://www.tensorflow.org/>
- [16] M. Feldman, «10 Real-World Examples of Machine Learning and AI [2018],» [En línea]. Available: <https://www.redpixie.com/blog/examples-of-machine-learning>
- [17] A. Martín, «Apuntes de transmisión del calor,» [En línea]. Available: <http://oa.upm.es/6935/1/amd-apuntes-transmision-calor.pdf>
- [18] «Active vs. Passive RDHx, White paper,» [En línea]. Available: <http://www.chilleddoor.com/files/uploads/2014/10/Cooling-Entire-Data-Centers-Using-Rear-Door-Heat-Exchangers-Motivair-White-Paper-1.pdf>
- [19] Data Center Knowledge, «Water cooled solutions for high density rack cooling,» [En línea]. Available: <http://www.datacenterknowledge.com/archives/2014/06/23/water-cooled-solutions-high-density-rack-cooling>
- [20] J. Park, «New Cooling Strategies for Greater Data Center Energy Efficiency, Facebook data center optimizacion,» *Facebook Data Centers*, [En línea]. Available: <https://code.facebook.com/posts/357772694354054/new-cooling-strategies-for-greater-data-center-energy-efficiency/>
- [21] T. Endo, A. Nukada, and S. Matsuoka, «TSUBAME-KFC: A modern liquid submersion cooling prototype towards exascale becoming the greenest supercomputer in the world,» *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 360–367, DOI: 10.1109/PADSW.2014.7097829 (2014).
- [22] J. T. Connor, R. D. Martin and L. E. Atlas, «Recurrent neural networks and robust time series prediction,» in *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 240-254, Mar 1994. doi: 10.1109/72.279188

- [23] M. Akram, C., «Sequence to Sequence Weather Forecasting with Long Short-Term Memory Recurrent Neural Networks,» *International Journal of Computer Applications* (0975 - 8887) Volume 143 - No.11, June 2016
- [24] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, «Recurrent Neural Networks for Multivariate Time Series with Missing Values,» *Nature Scientific Reports*, (2018) DOI:10.1038/s41598-018-24271-9
- [25] J. Xu, R. Rahmatizadeh, L. Bölöni and D. Turgut, «Real-Time Prediction of Taxi Demand Using Recurrent Neural Networks,» in *IEEE Transactions on Intelligent Transportation Systems*. doi: 10.1109/TITS.2017.2755684
- [26] B. Kermanshahi, «Recurrent neural network for forecasting next 10 years loads of nine Japanese utilities,» *Neurocomputing*, Volume 23, Issues 1–3, 1998, Pages 125-133, ISSN 0925-2312
- [27] «RaspberryPi, Raspbian,» [En línea]. Available: <https://www.raspberrypi.org/documentation/raspbian/>
- [28] «Raspberry Pi 3 Model B+ Datasheet,» [En línea]. Available: <https://static.raspberrypi.org/files/product-briefs/Raspberry-Pi-Model-Bplus-Product-Brief.pdf>
- [29] «UCI Machine Learning Repository. Beijing PM2.5 Data Data Set,» [En línea]. Available: <https://archive.ics.uci.edu/ml/datasets/Beijing+PM2.5+Data>
- [30] Reddy, V. S. and Shrestha Mohanty. «Deep Air : Forecasting Air Pollution in Beijing , China.» (2017). Semantic Scholar
- [31] D. Brodowski, «CPU frequency and voltage scaling code in the Linux(TM) kernel,» [En línea]. Available: <https://www.kernel.org/doc/Documentation/cpu-freq/user-guide.txt>
- [32] «Servidor Graphite GreenLSI,» [En línea]. Available: <http://visualizee.die.upm.es:8000/>
- [33] « Numpy and Scipy Documentation,» [En línea]. Available: <https://docs.scipy.org/doc/>
- [34] «Pandas: Python Data Analysis Library,» [En línea]. Available: <https://pandas.pydata.org/>
- [35] «Scikit-learn: Machine Learning in Python,» [En línea]. Available: <http://scikit-learn.org/stable/index.html>
- [36] «HDF5 for Python,» [En línea]. Available: <http://docs.h5py.org/en/stable/>
- [37] «Repositorio Github de ejemplos de keras,» [En línea]. Available: <https://github.com/keras-team/keras/tree/master/examples>
- [38] «Keras Documentation, Usage of optimizers,» [En línea]. Available: <https://keras.io/optimizers/>
- [39] «Keras Documentation, Usage of loss functions,» [En línea]. Available: <https://keras.io/losses/>

Anexo A: Aspectos éticos, económicos, sociales y ambientales

A.1 Introducción

Este proyecto se desarrolla en el contexto de un imparable crecimiento del desarrollo digital y la cantidad de población que vive en las ciudades. Esto conlleva numerosos retos energéticos, tecnológicos y organizativos, especialmente en el despliegue de nuevos centros de datos que atiendan la demanda computacional y de conectividad requerida.

El problema principal que implica a los centros de datos es la refrigeración, ya que las técnicas utilizadas hoy en día consumen casi un 40% de la energía total de la infraestructura y requieren de grandes dimensiones. Por tanto, se propone una solución disruptiva que permite reducir el impacto ambiental y energético, para permitir el despliegue de centros de datos en núcleos urbanos.

A.2 Descripción de impactos relevantes relacionados con el proyecto

El aspecto más importante de este proyecto es el energético, pues al tratarse de una refrigeración pasiva se consigue eliminar por completo el consumo de climatización. Además al no necesitar ningún tipo de ventilación también se aumenta exponencialmente la densidad de potencia de computación. De esta forma se pueden desplegar instalaciones muy eficientes y que ocupen poco espacio, lo que las hace unas muy buenas candidatas para su instalación en núcleos urbanos, y conlleva un ahorro económico destacable.

El despliegue de estos centros de datos permitirá un desarrollo sin inconvenientes de los dispositivos conectados en cuanto a cantidad y heterogeneidad. Además, utilizando aplicaciones como *Data Analytics* se puede aumentar en gran medida la eficiencia de los servicios de las ciudades del futuro. Todo ello implica un gran impacto social y ético en el desarrollo del conjunto de la sociedad y de las ciudades inteligentes.

También cabe destacar que el líquido dieléctrico utilizado, el *Novec 7100*, es respetuoso con el medio ambiente y la capa de ozono al contrario que sus predecesores en la industria (CFC, HFC, HCFC y PFC)

A.3 Análisis detallado de alguno de los principales impactos

En el apartado de introducción (Sección 1) se desarrollan más detalladamente los aspectos que se tratan en este anexo.

A.4 Conclusiones

Por todo lo descrito en este apartado podemos concluir que la solución propuesta implica un cambio de paradigma en el aspecto tecnológico, energético y económico del mercado de los centros de datos. Por tanto, su despliegue permitirá un importante desarrollo tecnológico y social de las ciudades inteligentes del futuro.

Anexo B: PRESUPUESTO ECONÓMICO

COSTE DE MANO DE OBRA

Horas	Precio/hora	Total
300	15 €	4.500 €

COSTE DE RECURSOS MATERIALES

	Precio de compra	Uso en meses	Amortización (en años)	Total
Ordenador personal <i>Lenovo Ideapad 700</i>	800,00 €	6	5	80,00 €
<i>Raspberry Pi 3 Model B+</i> (3 unidades)	120,00 €	6	4	15,00 €
Tarjeta de memoria microSD <i>Verbatim 16Gb</i> (3 unidades)	30,00 €	6	4	3,00 €
Líquido dieléctrico <i>Novec 7100</i> (6 kg)	300,00 €	6	4	37,50 €
Pecera de cristal	30,00 €	6	5	3,00 €
Alimentador 5V - 2,5A (3 unidades)	10,00 €	6	5	1,00 €
Cable de red <i>Ethernet</i> (3 metros)	5,00 €	6	5	0,50 €
<i>Switch</i> de conexión <i>Ethernet</i>	20,00 €	6	5	2,00 €
COSTE TOTAL DE RECURSOS MATERIALES				142,00 €

GASTOS GENERALES (costes indirectos)	15%	sobre CD	696,30 €
BENEFICIO INDUSTRIAL	6%	sobre CD+CI	320,30 €

MATERIAL FUNGIBLE

Impresión	100,00 €
Encuadernación	300,00 €

SUBTOTAL PRESUPUESTO	6.058,60 €
IVA APPLICABLE	21% 1.272,30 €

TOTAL PRESUPUESTO	7.330,90 €
--------------------------	-------------------