

UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Optimización en tiempo real del consumo de energía por  
ventiladores en un sistema de enfriamiento para un servidor  
por medio de redes neuronales**

Trabajo de graduación presentado por Josue David Valenzuela Herrera  
para optar al grado académico de Licenciado en Ingeniería en Ciencia de  
la Computación y Tecnologías de la Información

Guatemala,

2022



UNIVERSIDAD DEL VALLE DE GUATEMALA  
Facultad de Ingeniería



**Optimización en tiempo real del consumo de energía por ventiladores en un sistema de enfriamiento para un servidor por medio de redes neuronales**

Trabajo de graduación presentado por Josue David Valenzuela Herrera para optar al grado académico de Licenciado en Ingeniería en Ciencia de la Computación y Tecnologías de la Información

Guatemala,

2022



Vo.Bo.:



(f)

MSc. Douglas Barrios

Tribunal Examinador:



(f)

MSc. Douglas Barrios



(f)

Ing. Róger Artemio Díaz Fuentes

(f)

MSc. Juan Jose Celada Mazariegos

Fecha de aprobación: Guatemala, 12 de diciembre de 2022.



<b>Lista de figuras</b>	<b>VII</b>
<b>Lista de cuadros</b>	<b>IX</b>
<b>Resumen</b>	<b>XI</b>
<b>Abstract</b>	<b>XIII</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Antecedentes</b>	<b>3</b>
2.1. Optimal Self-Tuning PID Controller Based on Low Power Consumption for a Server Fan Cooling System . . . . .	3
2.2. Transforming Cooling Optimization for Green Data Center via Deep Reinfor- cement Learning . . . . .	3
<b>3. Justificación</b>	<b>5</b>
<b>4. Objetivos</b>	<b>7</b>
4.1. Objetivo general . . . . .	7
4.2. Objetivos específicos . . . . .	7
<b>5. Alcance</b>	<b>9</b>
<b>6. Marco teórico</b>	<b>11</b>
6.1. Teoría de control . . . . .	11
6.1.1. Control Loop . . . . .	11
6.1.2. Controlador PID . . . . .	12
6.1.3. Lead-lag compensators . . . . .	13
6.2. Sistemas de enfriamiento para servidores . . . . .	13
6.2.1. Enfriamiento por aire . . . . .	13
6.3. Redes neuronales . . . . .	14

<b>7. Diseño de framework de comunicación servidor-controlador</b>	<b>15</b>
7.1. Componentes del sistema . . . . .	15
7.2. API del sistema . . . . .	15
<b>8. Modelo de optimizacion de velocidad ventiladores</b>	<b>17</b>
<b>9. Diseño modelo pruebas</b>	<b>19</b>
9.1. Diseño modelo servidor-controlador . . . . .	19
9.2. Diseño experimental . . . . .	19
9.3. Resultados experimentación . . . . .	20
<b>10. Conclusiones</b>	<b>25</b>
<b>11. Recomendaciones</b>	<b>27</b>
<b>12. Bibliografía</b>	<b>29</b>
<b>13. Anexos</b>	<b>31</b>
13.1. Enlace repositorio framework de comunicación servidor-controlador . . . . .	31
13.2. Enlace repositorio implementacion de control PIDNN . . . . .	31
13.3. Intelligent Platform Management Interface (IPMI) Specification . . . . .	31
13.3.1. Baseboard Management Controller (BMC) . . . . .	32

---

## Lista de figuras

---

1. Estructura general del framework de comunicacion servidor-controlador . . . .	16
2. Diagrama de clases framework servidor-modelo . . . . .	16
3. Estructura red PIDNN . . . . .	17
4. Funciones de activación capa interna PIDNN . . . . .	18
5. ConFiguración de servidor y sistema de control velocidad ventilador . . . . .	20
6. ConFiguración interna de sistema de enfriamiento Dell Poweredge R210 . . . .	20
7. Diagrama de flujo auto ajuste ganancias PID por medio de PIDNN . . . . .	22
8. Resultados de control ganancias PID constantes (a) Temperatura del CPU (b) Ciclo de trabajo de ventiladores . . . . .	23
9. Resultados de control PIDNN: (a) Temperatura del CPU (b) Ciclo de trabajo de ventiladores . . . . .	23
10. Ganancias PID ajustados por PIDNN (a) Ganancia P (b) Ganancia I (c) Ganancia D . . . . .	24



---

Lista de cuadros

---

1. Resultados controladores . . . . . 21



El objetivo general del proyecto es la optimización del consumo de energía del sistema de enfriamiento en servidores IU por medio de la optimización de la velocidad de los ventiladores utilizados. Al mismo tiempo se busca desarrollar un framework de desarrollo que permita la fácil prueba e implementación de distintos modelos y métodos de optimización de los sistemas de enfriamiento. La optimización se logra mediante una red neuronal PIDNN la cual es entrenada en línea y ejecuta un algoritmo de control PID con el que se logra la optimización de la velocidad de los ventiladores.

Se encontró que la implementación de una red neuronal PIDNN puede reducir el consumo de energía hasta en un 11 %, alcanzando un estado estable durante el tiempo de respuesta de un servidor en estado estacionario ha estado de trabajo máximo en un lapso de tiempo de 200 segundos.



The general objective of the project is the optimization of the energy consumption of the cooling system in 1U servers by optimizing the speed of the fans used. At the same time, it seeks to develop a development framework that allows easy testing and implementation of different models and optimization methods for cooling systems. The optimization is achieved through a PIDNN neural network which is trained online and executes a PID control algorithm with which the fan speed optimization is achieved.

It was found that the implementation of a PIDNN neural network can reduce power consumption by up to 11 %, reaching a stable state during the response time of a server in steady state to maximum working state in a time span of 200 seconds. .



Uno de los grandes problemas modernos es el alto consumo de energía que es necesario para poder proveer servicios de tecnología de la información. El acelerado crecimiento de la demanda de servicios de tecnología de la información ha resultado en una alta demanda de servidores de nivel empresarial. Uno de los retos más grandes del desarrollo y administración de servidores es el enfriamiento de los equipos, los sistemas de enfriamiento son caracterizados por el alto consumo de energía que requieren para poder mantener la temperatura de un servidor como indicado por Kheirabadi et al. [1] en un data center el consumo de energía eléctrica por el sistema de enfriamiento puede llegar a ser de hasta el 33 % del total de energía utilizada. Los sistemas de enfriamiento comúnmente utilizados están basados en el uso de ventiladores por lo que la optimización del consumo de energía de los ventiladores se ha vuelto un punto de enfoque en el diseño de servidores modernos.

La gran mayoría de servidores modernos cuentan con múltiples procesadores y los componentes utilizados son almacenados con una alta densidad para poder alcanzar el desempeño que se necesita. Los servidores de alta densidad tienen una probabilidad de falla térmica más alta, por lo que requiere un mayor consumo de energía para el enfriamiento [2]. El consumo de energía solamente de los ventiladores puede llegar a ser hasta del 51 % del consumo total del sistema de enfriamiento [2], [3].

Este trabajo aspira a proponer un sistema de control automatizado apoyado por inteligencia artificial para la optimización de la velocidad de los ventiladores con el objetivo de reducir el consumo de energía del sistema de enfriamiento de un rack de servidores.



### **2.1. Optimal Self-Tuning PID Controller Based on Low Power Consumption for a Server Fan Cooling System**

Modelo de red neuronal PIDNN auto ajustable para la optimización del consumo de energía por medio de un modelo de energía de tercer orden para el consumo de energía de un ventilador en un sistema de simulación de un servidor 1U. [2]

### **2.2. Transforming Cooling Optimization for Green Data Center via Deep Reinforcement Learning**

Framework de aprendizaje de refuerzo profundo emergente para la optimización del consumo de energía en sistemas de enfriamientos para centros de datos. El modelo es entrenado para predecir el consumo de energía y los efectos en el sistema de enfriamiento con los que se alimenta una política de optimización en la configuración de control del sistema de enfriamiento. [4]



El crecimiento en la demanda de servidores de grado empresarial ha resultado en que el consumo de energía por data centers alcance entre el 1.12% y el 1.5% del total de energía eléctrica consumida en todo el mundo [5]. ASHRAE TC 9.9 define como esencial reducir el consumo de energía en sistemas de enfriamiento [6]. La necesidad de reducir el consumo de energía para poder enfriar un servidor ha llevado a diferentes estudios de técnicas, y optimizaciones de los sistemas de enfriamiento enfocadas en diferentes dimensiones de un servidor, que van desde una unidad 1U hasta un data center completo.

Debido a que el consumo de energía por ventiladores está directamente relacionado con la velocidad a la que funcionan el fin del estudio es proponer un modelo de controlador que regula de forma automática la velocidad de los ventiladores reduciendo el consumo de energía mientras se mantiene un nivel óptimo de rendimiento del sistema de enfriamiento para un rack de servidores 1U.



### 4.1. Objetivo general

Desarrollar un modelo de inteligencia artificial que optimice el consumo de energía en un sistema de enfriamiento de ventiladores para un servidor

### 4.2. Objetivos específicos

- Implementar un controlador auto-ajutable que reduzca el consumo de energía optimizando la velocidad de ventiladores en un sistema de enfriamiento para un rack de servidores 1U.
- Proveer un modelo de control que sea reproducible y abierto que pueda ser utilizado como comparación entre diferentes sistemas de optimización de sistemas de enfriamiento para servidores.



El proyecto se enfoca en el desarrollo de un modelo de optimización de consumo de energía de ventiladores en un sistema de enfriamiento por aire de un servidor 1U, para conseguirlo se cumpliría con:

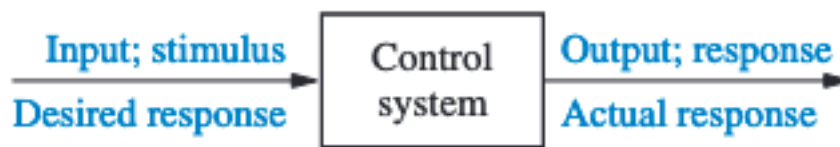
- Desarrollo de una red neuronal que optimice el consumo de energía de los ventiladores sin sacrificar el desempeño del sistema de enfriamiento.
- Diseño de un framework de software que facilite el desarrollo de nuevos modelos de optimización de sistemas de enfriamiento.



## 6.1. Teoría de control

Los sistemas de control tienen como objetivo obtener una salida, dada una entrada con un desempeño objetivo. Están formados por subsistemas y procesos que en conjunto dan como resultado la salida luego de recibir la entrada, las diferentes combinaciones de subsistemas y procesos determinan el desempeño del sistema.

El desempeño de los sistemas se mide en dos medidas principales: la respuesta transitoria y el error de estado estacionario. La respuesta transitoria se refiere a la respuesta del sistema al cambiar el estado luego de recibir una entrada. El error de estado estacionario es la diferencia entre la salida esperada y la salida arrojada por el sistema.

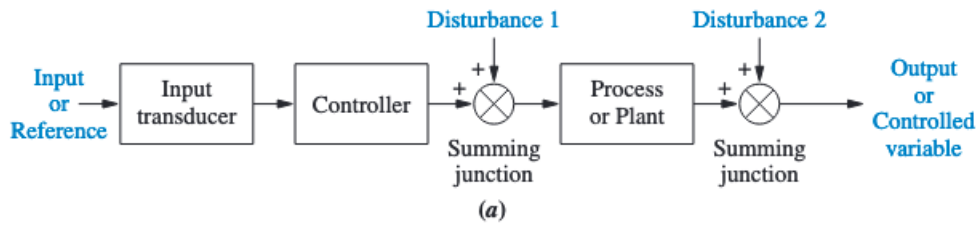


### 6.1.1. Control Loop

#### Sistemas Open-Loop

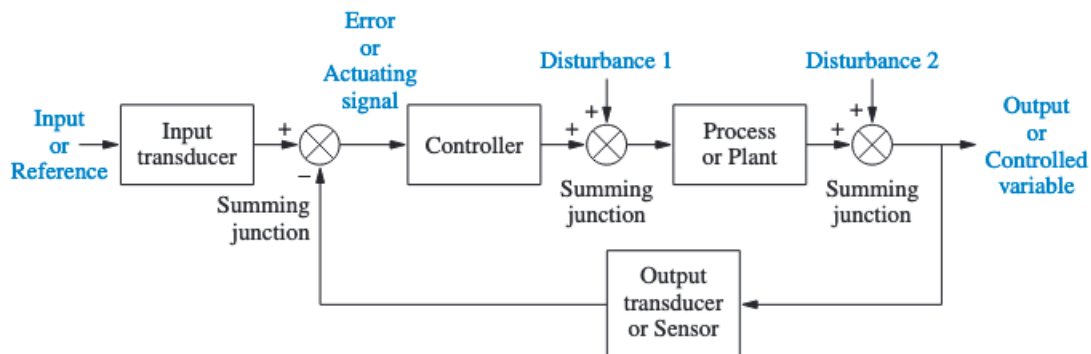
Un sistema Open-Loop consta de tres partes consecutivas: Subsistema de entrada, controlador, pCAPbibliografiaroceso. El subsistema de entrada es encargado de convertir la entrada a la forma que es utilizada en el controlador, el controlador se encarga de manejar el proceso. Entre cada una de las fases puede existir disturbios que afecten las señales que son transmitidas entre cada el controlador y el proceso, y en la salida del proceso. Una de las desventajas de los sistemas open-loop es que no son capaces de compensar cuando hay

disturbios en las señales. Estos sistemas basan su funcionamiento solamente en las entradas, por lo que no corrige ningún disturbio.



## Sistemas Closed-Loop

A diferencia de los sistemas Open-Loop, los sistemas Closed-Loop pueden compensar disturbios en las señales del sistema. Un sistema Closed-Loop consta de cinco partes consecutivas: Subsistema de entrada, Subsistema de error, Controlador, Proceso, Subsistema de Salida. El subsistema de entrada cumple con el mismo objetivo que para sistemas Open-Loop, convertir la entrada a la forma que es utilizada en el controlador. La entrada transformada es luego pasada al subsistema de error, el cual también tiene como entrada la señal del subsistema de salida, en este subsistema se calcula la señal de error como la diferencia entre la señal entrada y la señal de salida. Esta señal de error luego es transmitida al proceso, cuya salida es pasada al subsistema de salida. El cálculo de la medición de error da al sistema retroalimentación que permite corregir a posibles disturbios en las señales en cualquiera de los pasos.



### 6.1.2. Controlador PID

Es un método de control Closed-Loop con tres partes:

- Proporcional: Corresponde al presente. Trata de alcanzar el punto de referencia lo mas rápido posible.

- Integral: Corresponde al pasado. Trata de alcanzar el punto de referencia basado en la acumulación de errores en el pasado.
- Derivativa: Corresponde al futuro. Intenta predecir el error y llegar al punto de referencia con el error que se predijo.

El 90 % de los controladores utilizados en la industria son PID (o alguna variación), esto se debe a dos razones principales. Se pueden ajustar de forma heurística y no requieren un modelo de la planta exacto (que es lo más común en la práctica)

### **6.1.3. Lead-lag compensators**

Un compensador de adelanto-retraso es un componente de un sistema de control que mejora una respuesta de frecuencia no deseada en un sistema de retroalimentación y control.

Dada la planta de control, las especificaciones deseadas se pueden lograr utilizando compensadores. I, D, PI, PD y PID son controladores de optimización que se utilizan para mejorar los parámetros del sistema (como reducir el error de estado estable, reducir el pico resonante, mejorar la respuesta del sistema al reducir el tiempo de subida). Todas estas operaciones también pueden ser realizadas por compensadores, utilizados en la técnica de compensación en cascada.

## **6.2. Sistemas de enfriamiento para servidores**

El enfriamiento de la electrónica del servidor es una preocupación creciente para las empresas y las organizaciones gubernamentales involucradas en el procesamiento, el almacenamiento y las telecomunicaciones de datos. La base principal de esta preocupación es el costo y la complejidad cada vez mayores de la gestión térmica en las instalaciones de alojamiento de servidores. A medida que la industria continúa adoptando nuevas generaciones de procesadores que ofrecen capacidades y rendimiento computacional mejorados, el aumento del calor generado se vuelve inevitable.

En un centro de datos típico, aproximadamente el 33 % de la electricidad total consumida se asigna a la gestión térmica de la electrónica del servidor.

### **6.2.1. Enfriamiento por aire**

El enfriamiento por aire es actualmente la solución de enfriamiento mejor establecida adoptada por la industria del centro de datos y se implementa comúnmente a través del arreglo de pasillo frío/calor. El aire caliente de la instalación se enfría mediante el aire acondicionado de la sala de computadoras o las unidades de tratamiento de aire (CRAC/CRAH) y se envía a los pasillos fríos asignados a través de una cámara de aire debajo del piso. La diferencia aquí es que las unidades CRAH utilizan intercambiadores de calor agua-aire para enfriar el aire de la instalación, mientras que las unidades CRAC utilizan evaporadores llenos de refrigerante.

En el nivel del servidor, todos los componentes electrónicos se enfrían a través de convección de aire forzado directo con accesorios de disipador de calor necesarios para componentes de alto flujo de calor.

Los ventiladores de servidor dedicados impulsan el aire frío del pasillo a través de los servidores individuales y liberan el aire de escape calentado de regreso a la instalación a través de los pasillos calientes asignados. La eficiencia de este proceso se puede aumentar mediante el método de contención de pasillo; donde las corrientes de aire frío o caliente están confinadas para evitar la mezcla y recirculación del aire.

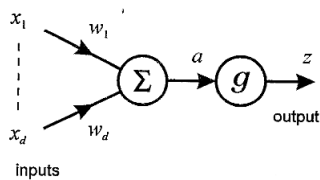
Debido a las malas propiedades termofísicas, el enfriamiento por aire generalmente requiere temperaturas de refrigerante más bajas y caudales más altos en relación con las estrategias de enfriamiento por líquido; ambos aumentan los gastos operativos.

### 6.3. Redes neuronales

Una red neuronal es un conjunto de nodos que intentan emular el comportamiento de las neuronas en el cerebro. En su forma más simple es un modelo que transforma una serie de variables de entrada en una salida. Las salidas se obtienen mediante operaciones matemáticas con diferentes parámetros de configuración, llamados pesos, que transforman las entradas en las salidas esperadas.

El reto de las redes neuronales es encontrar la combinación de pesos que, dada una entrada, obtengamos la salida deseada. El proceso para encontrar los pesos es llamado entrenamiento. 7

La estructura de una red neuronal tiene la siguiente forma:



Donde  $X_1$  y  $X_2$  son las entradas,  $w_i$  y  $w_d$  son los pesos y  $Z$  es la salida.

---

## Diseño de framework de comunicación servidor-controlador

---

Una de los elementos más importantes de los elementos más importantes para la correcta implementación de un modelo de optimización del sistema de optimización de los ventiladores es la comunicación entre el servidor y el sistema de optimización.

### 7.1. Componentes del sistema

El sistema de se divide en dos componentes importantes, el buffer de comunicación y el modelo de optimización. Ambos componentes interactúan entre sí para formar un ciclo de control cerrado. En la Figura 1 se muestra la interacción entre el buffer de comunicación y el modelo de optimización.

### 7.2. API del sistema

El framework implementa un ciclo de control cerrado para el cual es necesario agregar implementaciones del buffer de comunicación y el modelo de optimización. Para esto se brindan interfaces que al ser implementadas pueden ser intercambiadas, permitiendo así pruebas sencillas con diferentes modelos de optimización y diferentes estándares de comunicación con el servidor. La Figura 2 muestra la estructura de las interfaces y la comunicación entre sí.

Figura 1: Estructura general del framework de comunicacion servidor-controlador

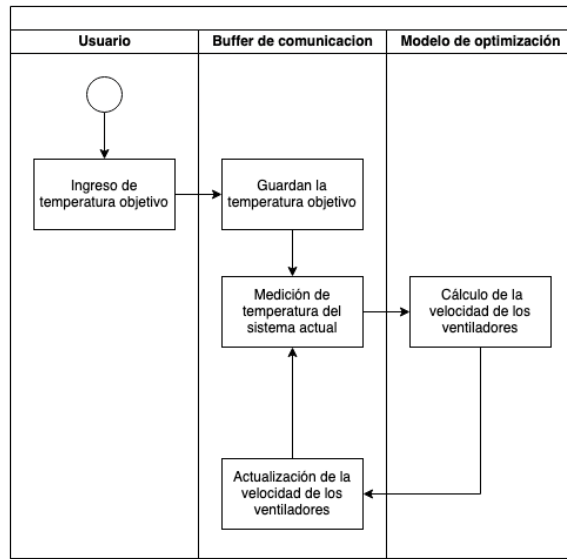
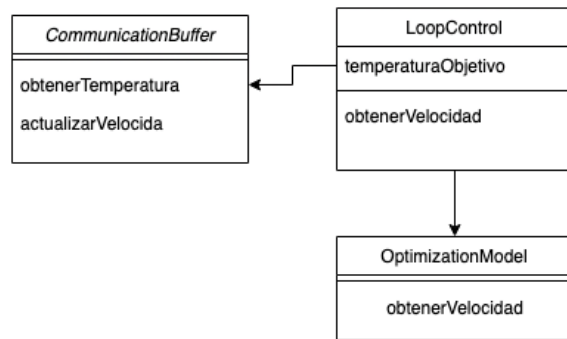


Figura 2: Diagrama de clases framework servidor-modelo



---

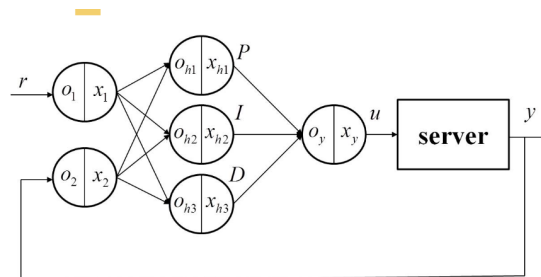
Modelo de optimización de velocidad ventiladores

---

Para el control de la velocidad de los ventiladores se utilizan controladores PID, estos debidos a la popularidad de uso y su extenso estudio en los parámetros óptimos de configuración. Se utilizará como punto de partida el modelo de optimización propuesto por Lee Chen [2] donde se utiliza un red neuronal PIDNN [8] para la optimización de las ganancias utilizadas en el controlador PID.

Una PIDNN es una red neuronal simple donde las capas están diseñadas para ejecutar el algoritmo de control. Es una estructura de 3 capas, ilustrada en la Figura 3. La primera capa es la capa de entrada donde se reciben la temperatura objetivo y la temperatura actual del sistema. La segunda capa tiene 3 neuronas las cuales actúan como las ganancias P, I y D. La tercera capa tiene una sola neurona que ejecuta el algoritmo de control y da salida a la nueva velocidad de los ventiladores.

Figura 3: Estructura red PIDNN



La red neuronal tiene pesos entre la capa de entrada y la capa intermedia constantes. Estos son  $W_{11} = W_{12} = W_{13} = 1$ , y  $W_{21} = W_{22} = W_{23} = -1$  esto ya que su objetivo es encontrar el error en el sistema, el cual está definido como la diferencia entre la temperatura objetivo y la temperatura actual del sistema.

La capa interna toma la señal de error enviada por la capa de entrada. La Figura 2 ilustra la relacion de cada neurona de la capa interna con las entradas enviadas por la capa de entrada de la red neuronal:

Figura 4: Funciones de activación capa interna PIDNN

$$x_{h1}(m) = \begin{cases} 1, & o_{h1}(m) > 1 \\ o_{h1}(m), & -1 \leq o_{h1}(m) \leq 1 \\ -1, & o_{h1}(m) < -1 \end{cases}$$

$$x_{h2}(m) = \begin{cases} 1, & o_{h2}(m) > 1 \\ o_{h2}(m) + o_{h2}(m-1), & -1 \leq o_{h2}(m) \leq 1 \\ -1, & o_{h2}(m) < -1 \end{cases}$$

$$x_{h3}(m) = \begin{cases} 1, & o_{h3}(m) > 1 \\ o_{h3}(m) - o_{h3}(m-1), & -1 \leq o_{h3}(m) \leq 1 \\ -1, & o_{h3}(m) < -1 \end{cases}$$

Como objetivo de la red neuronal, es la reducción del error en el sistema [7]

$$J = \sum_{t=1}^N E_t = \frac{1}{N} \sum_{t=1}^N (r(t) - y(t))^2$$

Donde  $r(t)$  es la temperatura objetivo y  $y(t)$  es la temperatura del sistema.

Los pesos entre la capa interna y la capa de salida, son encontrados utilizando las ecuaciones definidas por Lee Chen [2]

$$w_i(n+1) = w_i(n) - \eta \frac{\partial J}{\partial w_i}$$

$$\frac{\partial J}{\partial w_i} \approx -\frac{2}{N} \sum_{t=1}^N (r(t) - y(t)) \frac{y(t) - y(t-1)}{u(t) - u(t-1)} \dot{x}_i(t)$$

La capa de salida ejecuta el algoritmo de control PID, como fue definido por Lee Chen [2], para el modelo de un ventilador se puede encontrar con la ecuación:  $u(t) + Kp(y(t-1) - y(t)) + Ki(r(t) - y(t)) + Kd(-y(t) + 2y(t-1) - y(t-2))$  Donde  $Kp, Ki, Kd$  son las ganancias PID como salida de la capa escondida,  $u(t)$  es la velocidad del ventilador, y  $y(t)$  es la temperatura del sistema.

## 9.1. Diseño modelo servidor-controlador

El sistema de experimentación está compuesto por el sistema de control de velocidad de ventiladores y el servidor (Dell PowerEdge R210) al que se aplicará el sistema de control, ilustrado en la Figura 5. El servidor utiliza linux, con el que la herramienta IPMI y sensors se utilizan para modificar la velocidad de los ventiladores y medir la temperatura del CPU respectivamente. El sistema de control implementa el algoritmo de auto-ajuste y PID utilizando python instalado en una computadora utilizando MacOS. El sistema de control y el servidor se comunican por medio de una conexión ssh.

El sistema de control utiliza controladores PID para modificar la velocidad de los ventiladores y el sensor de temperatura del CPU brinda retroalimentación al controlador. Los ventiladores en el servidor ajustan la velocidad para subir o bajar el flujo de aire de entrada disminuyendo o bajando la temperatura de los componentes dentro del servidor. El servidor utiliza un sistema de enfriamiento basado en zonas, con 3 ventiladores, donde el ventilador 1 y 2 se encargan del enfriamiento del CPU y los slots de memoria, y el ventilador 3 en el enfriamiento del resto de componentes en la tarjeta madre, ilustrado en la Figura 6.

## 9.2. Diseño experimental

Para los fines del experimento se utilizará como punto de referencia la temperatura del CPU y se ajustará la velocidad de los ventiladores 1 y 2. La carga térmica es generada al aumentar el uso del CPU al 100%. Para elevar la carga sobre el CPU se utiliza la herramienta stress, la temperatura objetivo se configuró en 50 °C. El algoritmo de control requiere ganancias iniciales, estas deben de ser conFiguradas manualmente.

Figura 5: Configuración de servidor y sistema de control velocidad ventilador

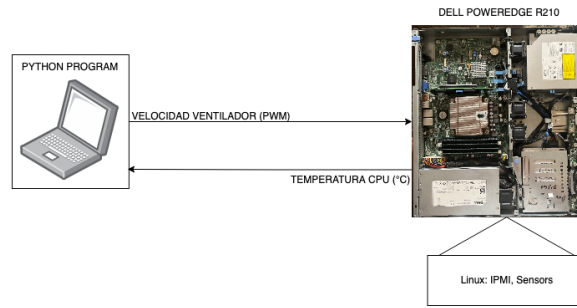
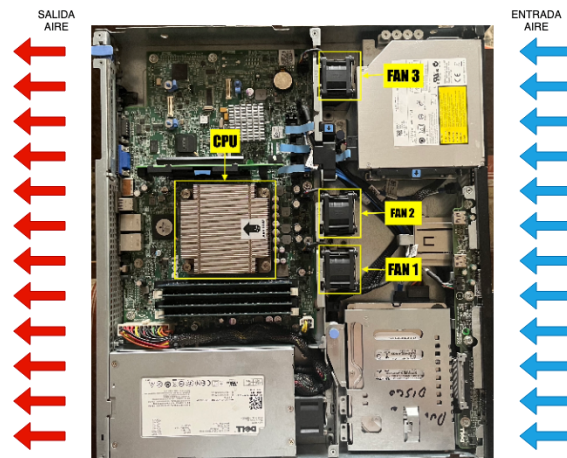


Figura 6: Configuración interna de sistema de enfriamiento Dell Poweredge R210



El diagrama de flujo de comunicación y ajuste entre el algoritmo de control y el servidor es ilustrado en la Figura 7.

El experimento tendrá una duración de 1000 segundos, esto para permitir que el sistema pueda alcanzar un estado estable y las mediciones de energía consumida tomen en cuenta el estado de subida y el estado estable.

### 9.3. Resultados experimentación

Las mediciones iniciales se realizaron utilizando ganancias PID constantes, estas ganancias se obtuvieron por medio de prueba y error. La temperatura, el ciclo de trabajo, y el consumo de energía de los ventiladores obtenidos utilizando ganancias PID constantes se muestran en la Figura 5. El proceso es repetido utilizando el algoritmo para obtener ganancias PID, las ganancias iniciales se utilizan los mismos valores utilizados para el controlador PID común, estas son ajustadas automáticamente por la red neuronal durante el tiempo de medición, las ganancias PID se muestran en la Figura 9, estas ganancias son actualizadas en el controlador PID en tiempo real para calcular la velocidad de los ventiladores, los resultados de temperatura, y el ciclo de trabajo de los ventiladores obtenidos se muestran en la Figura 8.

El auto ajuste de las ganancias PID por medio de PIDNN tardan alrededor de 200 segundos en alcanzar el estado estable, la Figura 9 ilustra las ganancias P, I y D a través del tiempo. La temperatura objetivo es alcanzada en alrededor de 400 segundos antes en el controlador auto-ajutable comparado con el controlador con ganancias constantes, además el overshoot es menor, el Cuadro 1 muestra los resultados obtenidos con ambos controladores.

Cuadro 1: Overshoot y consumo de energía bajo controladores PID y PIDNN

Controlador	overshoot	Consumo de energía ( $W$ )
PID	22.0 %	107.24
PIDNN	11.5 %	98.42

Durante el intervalo de pruebas de 1000 segundos el modelo de auto ajuste propuesto redujo el consumo de energía en un 11 % para un servidor 1U que pasa de un estado idle a un estado de trabajo máximo.

Figura 7: Diagrama de flujo auto ajuste ganancias PID por medio de PIDNN

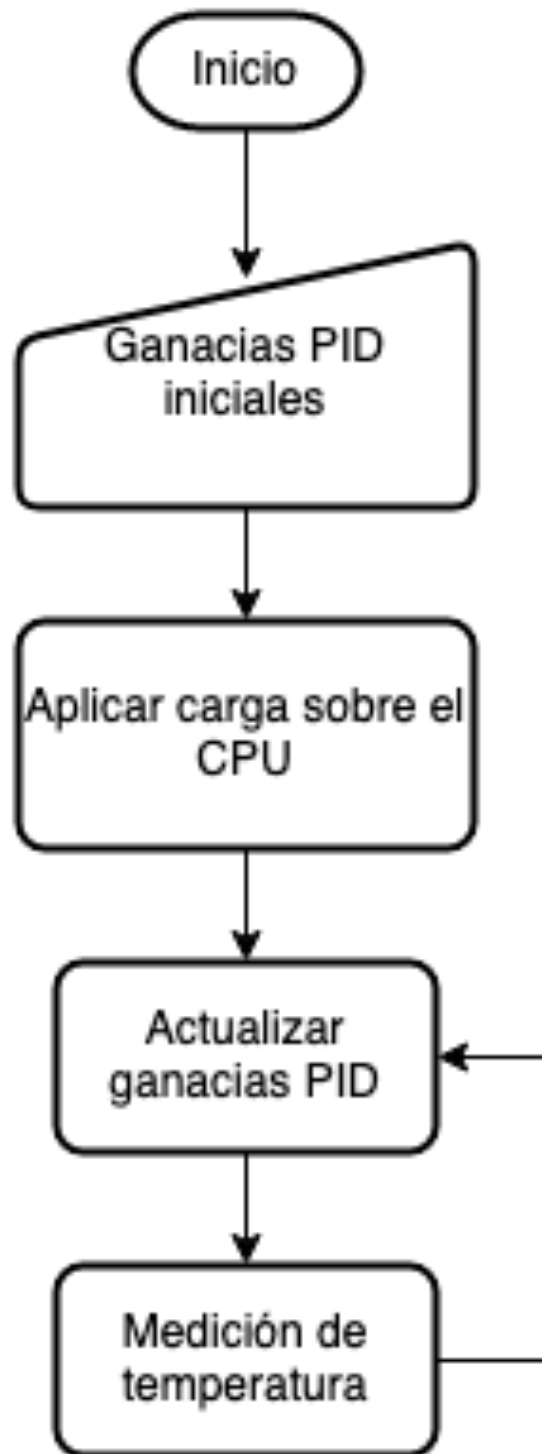


Figura 8: Resultados de control ganancias PID constantes (a) Temperatura del CPU (b) Ciclo de trabajo de ventiladores

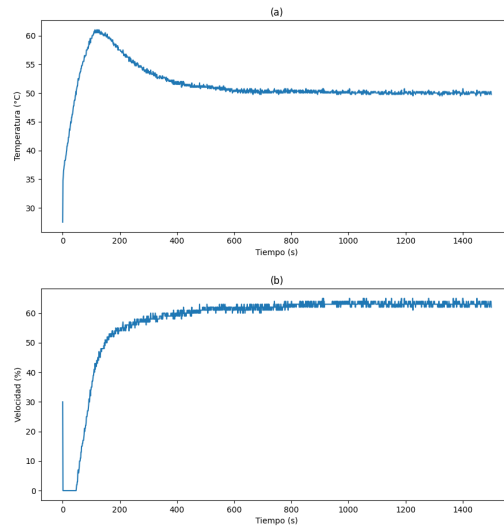


Figura 9: Resultados de control PIDNN: (a) Temperatura del CPU (b) Ciclo de trabajo de ventiladores

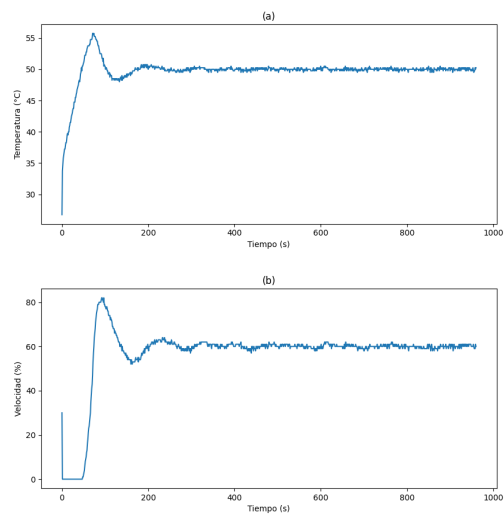
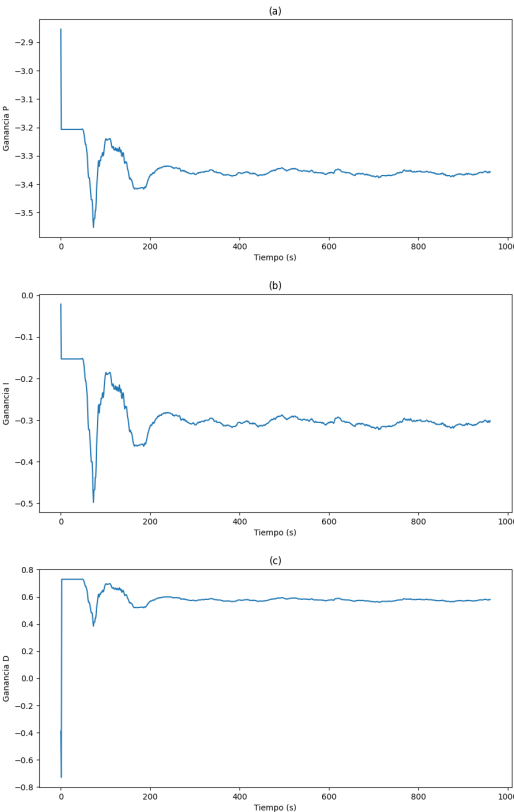


Figura 10: Ganancias PID ajustados por PIDNN (a) Ganancia P (b) Ganancia I (c) Ganancia D



- El modelo propuesto para el auto ajuste de velocidad de ventiladores del sistema de enfriamiento de un servidor 1U logró exitosamente reducir el consumo de energía de forma significativa, apoyándose en conceptos de control existentes.
- Durante un periodo de 1000 segundos hasta un 11 % del consumo de energía puede ser reducido, y además reducir el overshoot hasta un 10 % por medio del auto ajuste de ganancias PID.
- El framework diseñado para la supervisión y control del sistema de enfriamiento en servidores que implementan el estándar IPMI permiten la implementación de y comparación de algoritmos de control para optimizar el sistema de enfriamiento.



# CAPÍTULO 11

---

## Recomendaciones

---

Uno de los limitantes en la optimización del modelo propuesto es la precisión de los sensores de temperatura en el CPU, esto debido a la relación entre la capa interna y la capa de salida. Los sensores de temperatura tienen una precisión de  $0.25\text{ }^{\circ}\text{C}$  lo que limita la optimización de las ganancias PID. Utilizar un sensor con mayor precisión ayuda a mejorar la optimización de los pesos en la red neuronal.

Para servidores con sistemas de enfriamiento por zonas utilizar sensores de temperatura por cada zona permite la optimización de cada zona independientemente de las otras, esto resulta en reducción del consumo de energía.



- 
- 
- [1] A. C. Kheirabadi y D. Groulx, “Cooling of Server Electronics: A design review of existing technology,” *Applied Thermal Engineering*, vol. 105, págs. 622-638, 2016. DOI: [10.1016/j.applthermaleng.2016.03.056](https://doi.org/10.1016/j.applthermaleng.2016.03.056).
  - [2] C. Lee y R. Chen, “Optimal self-tuning PID controller based on low power consumption for a server fan cooling system,” *Sensors*, vol. 15, n.º 5, págs. 11 685-11 700, 2015. DOI: [10.3390/s150511685](https://doi.org/10.3390/s150511685).
  - [3] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler y T. Keller, “Energy Management for commercial servers,” *Computer*, vol. 36, n.º 12, págs. 39-48, 2003. DOI: [10.1109/mc.2003.1250880](https://doi.org/10.1109/mc.2003.1250880).
  - [4] Y. Li, Y. Wen, D. Tao y K. Guan, “Transforming cooling optimization for Green Data Center via deep reinforcement learning,” *IEEE Transactions on Cybernetics*, vol. 50, n.º 5, págs. 2002-2013, 2020. DOI: [10.1109/tcyb.2019.2927410](https://doi.org/10.1109/tcyb.2019.2927410).
  - [5] S.-W. Ham, M.-H. Kim, B.-N. Choi y J.-W. Jeong, “Simplified server model to simulate data center cooling energy consumption,” *Energy and Buildings*, vol. 86, págs. 328-339, 2015. DOI: [10.1016/j.enbuild.2014.10.058](https://doi.org/10.1016/j.enbuild.2014.10.058).
  - [6] En *Thermal guidelines for Data Processing Environments*. American Society of Heating, Refrigerating, y Air-Conditioning Engineers, 2009, págs. 205-205.
  - [7] C. M. Bishop, “Neural networks and their applications,” *Review of Scientific Instruments*, vol. 65, n.º 6, págs. 1803-1832, 1994. DOI: [10.1063/1.1144830](https://doi.org/10.1063/1.1144830).
  - [8] H. Shu e Y. Pi, “PID neural networks for time-delay systems,” *Computers (E) Chemical Engineering*, vol. 24, n.º 2-7, págs. 859-862, 2000. DOI: [10.1016/s0098-1354\(00\)00340-9](https://doi.org/10.1016/s0098-1354(00)00340-9).
  - [9] Intel, DELL, Hewlett-Packard y NEC, “Intelligent Platform Management Interface Specification Second Generation,” *A Joint Message from the IPMI Promoters*, 2013. dirección: <https://www.intel.com/content/www/us/en/products/docs/servers/ipmi/ipmi-home.html>.



## **13.1. Enlace repositorio framework de comunicación servidor-controlador**

<https://github.com/jdvalenzuelah/sfc>

## **13.2. Enlace repositorio implementacion de control PIDNN**

<https://github.com/jdvalenzuelah/smart-fancontrol>

## **13.3. Intelligent Platform Management Interface (IPMI) Specification**

La especificación IPMI define una interfaz estandarizada para la administración de plataformas de hardware. Incluye interfaces para el monitoreo y control de las funciones del hardware, principalmente utilizadas para determinar la salud de la plataforma.

IPMI define interfaces para el monitoreo de temperatura, voltaje, ventiladores, fuentes de poder, errores de bus, checks de seguridad físicos. Incluye recuperación de errores de forma remota o local y registro de condiciones anormales.

La especificación define la arquitectura, comandos comunes, eventos y formatos de almacenamiento de registros.

### 13.3.1. Baseboard Management Controller (BMC)

Se define como la implementación centralizada de las funciones más comunes de la especificación IPMI. En algunos sistemas el BMC puede ser el único controlador. [9](#)

- Generalmente un BMC implementa las funciones:
- Interfaz del sistema: Comunicación entre el software del sistema y el BMC
- Controlador de mensajes: Ruteo de mensajes
- Interfaz System Event Log (SEL): Registro de eventos
- Generador de eventos: Monitoreo y transmisión de eventos.
- Repositorio Sensor Data Record (SDR): Acceso a datos generados por los sensores del equipo.
- Comandos OEM: Comandos específicos a la implementación de IPMI definidos por el OEM.