



**INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES  
DE MONTERREY**

**TC1031 - Programación de estructuras de datos y algoritmos fundamentales**

Profesor: David Sánchez

**Módulo 2: Balanceador de Carga (Round Robin)**

Grupo 604

Diana María Arámburo Lozano | A01646337

Camila Gomez Godinez | A01639319

Gabriela Ruelas Gaytán | A01640880

Emilio Guzmán Flores | A01643485

Samantha Mailen Gallardo Mota | A01640886

*Lunes 1 de diciembre del 2025*

## **Módulo 2: Balanceador de Carga (Round Robin)**

### **Descripción del algoritmo Round Robin**

El algoritmo Round Robin es una estrategia de asignación que se caracteriza por distribuir tareas de forma cíclica y equitativa entre un conjunto de recursos disponibles. En el contexto de un balanceador de carga, este algoritmo asigna cada solicitud entrante a los servidores siguiendo un orden secuencial fijo, regresando al primer servidor una vez que se ha llegado al último, generando así un ciclo continuo.

Su funcionamiento se basa en un puntero que avanza de servidor en servidor conforme llegan las solicitudes. Cada request es enviada al servidor actual, y posteriormente el puntero se mueve al siguiente nodo en la secuencia. Este proceso se repite indefinidamente, garantizando que todos los servidores reciban una cantidad similar de solicitudes sin considerar factores adicionales como el tiempo de respuesta o la carga real del servidor.

La principal ventaja del Round Robin es su simplicidad y predictibilidad, ya que no requiere cálculos complejos ni monitoreo constante del estado de los servidores. Todos los recursos participan de manera equitativa, haciendo que sea una solución eficiente en escenarios donde los servidores tienen capacidades similares.

### **Justificación del modelo implementado**

El modelo implementado utiliza una lista enlazada circular para representar el conjunto de servidores disponibles, lo cual se alinea de forma natural con la lógica del algoritmo Round Robin. La estructura circular permite recorrer los servidores de forma continua sin necesidad de reiniciar manualmente el índice cuando se llega al final, ya que el último servidor apunta nuevamente al primero.

Esta elección mejora la eficiencia del diseño al evitar comprobaciones adicionales sobre los límites del arreglo, simplifica el control del flujo y refleja el comportamiento real de un sistema de balanceo. Además, la lista enlazada circular permite escalabilidad, ya que sería sencillo añadir o eliminar servidores dinámicamente sin necesidad de reorganizar toda la estructura, como ocurriría con un arreglo estático.

La implementación también incorpora un conteo individual de solicitudes por servidor, lo cual permite generar estadísticas claras de desempeño y una visualización de la distribución de carga, facilitando el análisis del comportamiento del sistema y su correcta operación. En conjunto, este modelo combina claridad conceptual, eficiencia operacional y facilidad de mantenimiento, resultando adecuado para simulaciones educativas y sistemas básicos de balanceo de carga.

### **Complejidad del proceso de asignación**

El proceso de asignación mediante Round Robin tiene una complejidad temporal de:  $O(n)$ , donde  $n$  representa el número total de solicitudes.

Esto se debe a que cada request se procesa exactamente una vez y la operación realizada en cada asignación consiste únicamente en incrementar un contador y avanzar el puntero al siguiente servidor, ambos procesos de tiempo constante.

Por lo tanto, la operación individual de asignación tiene complejidad  $O(1)$ , y al repetirse para cada solicitud, el tiempo total escala linealmente con la cantidad de requests.

## Referencias

GeeksforGeeks. (2025, 22 septiembre). *Load balancing algorithms*. GeeksforGeeks.

<https://www.geeksforgeeks.org/system-design/load-balancing-algorithms/>

GeeksforGeeks. (2025, septiembre 15). *Introduction to Circular Linked List*. GeeksforGeeks.

<https://www.geeksforgeeks.org/dsa/circular-linked-list/>