

# Gestión de procesos basado en lógica difusa con estrictos niveles de consenso

Logically diffused process management with strict levels of consensus

Jorge T. Fornerón Martínez<sup>1</sup>, Federico Agostini<sup>2</sup>, David L. La Red Martínez<sup>2</sup>

<sup>1</sup> Universidad Nacional de Pilar, Paraguay

<sup>2</sup> Universidad Nacional del Nordeste, Argentina

jforneron@aplicadas.edu.py , agostinifede@hotmail.com , lrmdavid@exa.unne.edu.ar

**RESUMEN.** El funcionamiento de los sistemas distribuidos obliga a que los grupos de procesos deban tomar decisiones en base a acuerdos en relación con el acceso a los recursos. Esas decisiones pueden estar relacionadas al hecho de compartir o no recursos y con el nivel de consenso o acuerdo necesario. La creación de un nuevo modelo de decisión es una propuesta interesante, con la funcionalidad de adaptarse a distintos requerimientos y que los procesos tengan la posibilidad de acceder a los recursos compartidos mediante la exclusión mutua con exigencias estrictas de consenso para asignar los recursos de manera consecutiva cuando son solicitados por un proceso o grupo de procesos. En cada nodo es definida una interfaz entre las aplicaciones y el sistema operativo, que mediante un runtime (software en tiempo de ejecución) incluido en la interfaz mencionada, gestiona tanto los procesos como los recursos compartidos y define el escenario correspondiente. Para el intercambio de información, los runtime interactúan entre sí y donde en uno de los nodos existe un runtime coordinador global cuya función es evaluar y ejecutar el modelo de decisión y el operador de agregación correspondiente. En el presente documento, se propone un operador de agregación que tiene como fin favorecer la asignación de los recursos a los procesos. El operador de agregación permite la toma de decisiones en la gestión de grupos de procesos. Esto se realiza mediante la autorregulación y la utilización de etiquetas lingüísticas que permitan clasificar los datos del estado del sistema, además de nodos, recursos y procesos.

**ABSTRACT.** The operation of distributed systems requires that process groups must make decisions based on agreements regarding access to resources. These decisions may relate to share or not share resources and to the level of consensus or agreement required. The creation of a new decision model is an interesting proposal, with the functionality to adapt to different requirements and that the processes have the possibility to access shared resources through mutual exclusion with strict consensus requirements to allocate resources consecutively when they are requested by a process or group of processes. In each node a interface is defined between the applications and the operating system, by executing a runtime included in the mentioned interface, manages both the processes and the shared resources and defines the corresponding scenario. For the exchange of information, the runtimes interact with each other and where in one of the nodes there is a global coordinating runtime whose function is to evaluate and execute the decision model and the corresponding aggregation operator. In this document, an aggregation operator is proposed that is intended to favor the assignment of resources to processes. The aggregation operator enables decision making in process group management. This is done through self-regulation and the use of language tags that allow the classification of system status data, as well as nodes, resources and processes.

**PALABRAS CLAVE:** Operadores de agregación, Control de concurrencia, Comunicación entre grupos de procesos, Exclusión mutua, Sistemas operativos, Programación de procesadores, Lógica difusa, Computación con palabras.

**KEYWORDS:** Aggregation operators, Concurrency control, Communication between groups of processes, Mutual exclusion, Operating systems, Processor scheduling, Fuzzy logic, Computing with word.



## 1. Introducción

El concepto de empresa virtual está basado en la capacidad de crear cooperaciones entre empresas para captar oportunidades de negocios que por sí mismos no lo podrían realizar. Dicho de otra manera, la cooperación es la clave para proporcionar una nueva solución de manera conjunta para aprovechar oportunidades imprevistas.

La empresa virtual tiene características particulares como la agilidad, creación de valor a partir de oportunidades o la capacidad de identificar esas oportunidades, además de constituirse en una forma alternativa de diferenciar e integrar el trabajo en condiciones dinámicas, a través de la cooperación, según se menciona en el trabajo de Lau, Chin, Pun y Ning (2000).

La formación Redes de Empresas Virtuales o Virtual Enterprise Network (VEN) está tomando impulso para explorar nuevas oportunidades de negocios. En general, las empresas virtuales pueden definirse como una forma de organizar actividades comerciales, donde diferentes socios independientes explotan la oportunidad de negocio mediante la cooperación empresarial. La evaluación continua del desempeño de las empresas asociadas para un determinado tipo de tarea comercial es un factor importante para el éxito de éstas y el apoyo en la toma de decisiones es una funcionalidad básica para el rendimiento óptimo de las VEN, de acuerdo a lo expresado por Katzy y Schuh (1998).

Las VEN se apoyan en la utilización de tecnología actual para lograr su objetivo. La utilización de la inteligencia artificial o la minería de datos son un claro ejemplo. El Procesamiento Analítico en Línea - On-Line Analytical Process (OLAP), que se basa en la tecnología de minería de datos, ha sido desarrollado por empresas de software y productos comerciales, como el servidor OLAP de Microsoft.

En los últimos años se han producido cambios significativos en lo que respecta a los paradigmas de los procesos de fabricación, pasando del modelo tradicional al modelo de fabricación ágil y virtual, aprovechando los avances de las tecnologías de la información y la red Internet. Los beneficios de la adopción de las tecnologías de Internet pueden demostrarse mediante el mejoramiento de las actividades de fabricación en diversos aspectos de las actividades de la cadena de valor, desde el diseño de productos y procesos hasta la comercialización, de acuerdo a Dell y Fredman (1999).

Esta actividad de fabricación es la parte de la economía que produce bienes materiales altamente mecanizados y automatizados. Desde el comienzo de la industrialización, los saltos tecnológicos han dado lugar a cambios de paradigma, denominados "revoluciones industriales".

Un nuevo cambio de paradigma en la producción industrial se plantea a partir de una digitalización avanzada dentro de las fábricas, la combinación de las tecnologías de Internet y las tecnologías orientadas al futuro en el campo de los objetos "inteligentes" (máquinas y productos).

A ese cambio de paradigma (4ta. revolución industrial) se lo conoce más comúnmente por Industria 4.0, término asociado a las versiones de software. Lasi, Fettke, Kemper, Feld y Hoffmann (2014).

Actualmente conceptos como Internet de las Cosas (Internet of Things - IOT) son comunes en la Industria 4.0, la cual incorpora tecnología inalámbrica, sensores y máquinas inteligentes para crear una empresa conectada. La Internet Industrial de las Cosas (Industrial Internet of Things - IIoT) se considera un componente significativo de la Industria 4.0; recoge información sobre procesos, rendimiento de los equipos, suministros y pedidos, y usando grandes datos, agrega información de proveedores, fabricantes y clientes, según lo mencionado por Khan, Byun y Park (2020).

El rápido desarrollo de la tecnología eléctrica y electrónica, la tecnología de la información y la tecnología de fabricación avanzada, el modo de producción de las empresas manufactureras permite que se cambie de lo



digital a lo inteligente. La combinación de la tecnología de realidad virtual basada en el Sistema Ciberfísico (Cyber Physical System - CPS) está incorporándose a las industrias. La tecnología de fabricación inteligente es una de las áreas de alta tecnología a las que los países industrializados prestan más atención. Se han propuesto en consecuencia estrategias en Europa, China y Estados Unidos.

En el contexto de la fabricación inteligente, es importante establecer la fábrica inteligente para lograr una fabricación avanzada basada en tecnologías de red y datos del proceso de fabricación. Además, la implantación de la fábrica inteligente debe tener en cuenta el statu quo y los requisitos de fabricación. Debido a las diferentes características del campo de la fabricación y del campo de la información, deber resolverse aún muchos problemas técnicos para acelerar la implementación de la fábrica inteligente, de acuerdo con Chen et al. (2018).

Wan, Cai y Zhou (2015) refieren que la Industria 4.0 fue propuesta y lanzada en 2013. La integración de la tecnología de fabricación, la tecnología digital y la tecnología de redes puede aplicarse ahora al servicio de gestión del diseño y la producción. Además, el proceso de fabricación posee ahora las características (por ejemplo, la percepción, el análisis, la decisión y el control) necesarias para satisfacer los requisitos de respuesta dinámica del producto, así como el rápido desarrollo de nuevos productos.

Se tiene un nuevo escenario donde se observa la competencia en el entorno empresarial y con los antecedentes técnicos mencionados, las industrias manufactureras mundiales modernas y las tecnologías de la información están profundizando sus grados de integración. Tecnologías como la IOT, con infraestructura de bases de datos distribuidas, Computación en la Nube (Cloud Computing), los grandes datos (Big Data), la computación móvil, las redes industriales inalámbricas y la tecnología incrustada (embedded technology) son términos comunes en este escenario. Este progreso también trae consigo una mayor esperanza de nuevas aplicaciones, como la personalización de la producción y la gestión del ciclo de vida de los productos, según menciona Wan et al. (2016).

El "Big data" es una enorme colección de datos de gran tamaño y volumen, generalmente Exabyte; almacenados en granjas de datos. En los últimos decenios se ha demostrado que las grandes empresas comerciales dispersas en numerosas zonas geográficas remotas han practicado la acumulación de historiales de sus clientes en relación con las ventas, de diferentes empresas sociales y sitios web de comercio y negocios electrónicos en diversos formatos necesarios en los que los científicos de los datos tratan de explorar y correlacionar los conjuntos de datos para realizar y publicar predicciones valiosas para la adopción oportuna de decisiones para las perspectivas de las empresas con la mitigación de los riesgos.

Hoy, el e-business (comercio electrónico) está en las manos de toda persona que utilice un teléfono móvil o tenga acceso a una computadora, donde atender llamadas, enviar y recibir mensajes de WhatsApp, compartir llamadas de voz y vídeo, realizar reservas o comprar diversos artículos es la rutina, apoyada en la extraordinaria revolución en el campo de la tecnología informática y de las comunicaciones - la www (World Wide Web) – donde las principales empresas invirtieron considerables presupuestos para adoptar esta tecnología.

Las tecnologías básicas de la informática, es decir, la inteligencia artificial, la informática distribuida, las estructuras de red, el aprendizaje automático y la inteligencia empresarial, son útiles para la evolución de la red social. "Business Intelligence (BI)" la palabra acuñada por primera vez en 1989, proporciona poder de decisión a los científicos de datos, gerentes de oficina, ejecutivos corporativos y clientes relacionados para tomar una decisión genuina sobre la compra y venta de negocios con la existencia de herramientas basadas en la tecnología de sistemas distribuidos para analizar instantáneamente los datos y predecir los primeros resultados, según Saleem et al. (2019).

Los sistemas distribuidos, compuestos de múltiples nodos y múltiples procesos, que de manera cooperativa realizan una determinada función, requieren la utilización de modelos de decisión que manera que permitan



la utilización de los recursos compartidos a los grupos de procesos que los requieran, accedidos mediante la modalidad de exclusión mutua.

Soluciones propuestas para este problema se tienen en Tanenbaum (1996) y en Tanenbaum (2009), donde son descritos los principales algoritmos de sincronización en sistemas distribuidos. Agrawal y El Abbadi (1991) presentan una eficiente solución, también tolerante a fallos para el problema de la exclusión mutua distribuida. Ricart y Agrawala (1981), Cao y Shingal (2001) y Lodha y Kshemkalyani (2000) presentan algoritmos para la gestión de la exclusión mutua en redes informáticas. Stallings (2005) se enfoca en los principales algoritmos para la gestión de procesos distribuidos, exclusión mutua distribuida y estados globales distribuidos.

En los sistemas distribuidos, la asignación de recursos a los procesos debe ser realizada considerando las prioridades de los procesos y el estado de la carga de trabajo de los nodos computacionales en los cuales se ejecutan los procesos.

Además, se han propuesto soluciones que podríamos llamar clásicas para diversos tipos de sistemas distribuidos en Andrews (2000), Guerraoui y Rodríguez (2006), Lynch (2000), Tel (2000) y en Attiya y Welch (2004). También en Saxena y Rai (2003) y en Velázquez (1993) se presentan trabajos centrados en asegurar la exclusión mutua. Lin, Lian, Chen y Zhang (2004) presentan una interesante solución distribuida basada en permisos y en Sha, Rajkumar y Lehoczky (1990) una solución basada en prioridades de proceso. Una interesante propuesta se presentó en Ríos y La Red Martínez (2019).

Como se mencionó en Wen, Chang y Lai (2020), sobre la base del concepto de traducción simbólica, Herrera y Martínez propusieron el modelo de representación lingüística 2-tupla, que expresa la información de evaluación lingüística usando la 2-tupla lingüística  $(si, \alpha)$ , donde el elemento semántico  $si$  es una etiqueta lingüística de una variable lingüística predefinida,  $s$ , y  $\alpha$  es un valor numérico que representa la traducción simbólica. Además, Zhang et al. (2015) introdujeron los conjuntos imprecisos difusos dudosos y valorado a intervalos, combinando el modelo de conjuntos difusos dudosos y el de conjuntos imprecisos valorado a intervalos. Tao y otros (2015) presentaron el método de conjunto impreciso 2-tupla, que incorpora el conjunto de términos lingüísticos 2-tupla y el conjunto impreciso para resolver complejos problemas de toma de decisiones en grupo.

Hoy en día, los métodos del conjunto incompleto se aplican ampliamente para resolver problemas de adopción de decisiones en la vida real (por ejemplo, Ali y Shabir; Chang; Deli y Cagman; Tang; Chang; Chang y otros), citados en Wen et al. (2020).

Diferentes palabras pueden tener un significado diferente para diferentes personas. Las respuestas sobre cómo se puede validar un motor de Computación con Palabras (CWW), qué modelos de conjuntos difusos se deben utilizar o qué opciones se deben tomar para mantener el diseño del motor de CWW lo más simple posible, se analizan y fundan en Mendel (2007).

Los nuevos modelos de decisión para la asignación de recursos compartidos podrían ejecutarse en el contexto de un gestor de recursos compartidos para el sistema distribuido, que recibiría los requisitos de recursos compartidos de los procesos que se ejecutan en los diferentes nodos distribuidos, así como el estado de carga computacional de los nodos.

Se ha trabajado con variables borrosas utilizando etiquetas lingüísticas y 2-tuplas para evitar la pérdida de precisión en el cálculo con palabras.

Un modelo similar se ha presentado en Herrera y Martínez (2000), Zapata, Fuentealba y Valenzuela (2015), Jiménez y Zulueta (2017), Ying (2002) y Liu, Yin y Pei (2018). Estos trabajos muestran diferentes ventajas de este formalismo para representar la información lingüística sobre los modelos clásicos.



Se trata de un dominio lingüístico continuo, en el que el modelo de cálculo lingüístico se basa en tuplas lingüísticas y realiza los procesos de cálculo de palabras con facilidad y sin pérdida de información, por lo que los resultados de los procesos de cálculo de palabras se expresan siempre en el dominio lingüístico inicial.

Debido a estas ventajas, este modelo de representación lingüística se utilizará para lograr el desarrollo de un procedimiento de fusión de la información lingüística y numérica.

El modelo de representación lingüística de 2-tuplas representa la información lingüística mediante una 2-tupla,  $(s, \alpha)$ . En este trabajo se utilizará para representar la carga de los nodos, las preferencias nodales y las prioridades finales.

La traducción simbólica de un término lingüístico  $s_i \in S = \{s_0, \dots, s_g\}$  consiste en un valor numérico  $\alpha_i \in [-.5, .5]$  que soporta la "diferencia de información" entre una cuenta de información  $\beta$  evaluada en  $[0, g]$  obtenida después de una operación de agregación simbólica (actuando sobre el índice de orden de las etiquetas) y el valor más cercano en  $[0, \dots, g]$  que indica el índice del término lingüístico más cercano en  $S (s_j)$ .

En el presente documento se presentará un nuevo operador de agregación específico para el problema mencionado. Este operador entra dentro de la categoría de operadores de OWA, más específicamente de Neat OWA. Se presentará un método innovador para la gestión de recursos compartidos en sistemas distribuidos.

## 2. Estructura de datos

El escenario propuesto considera las siguientes condiciones. En primer lugar, los procesos deben tener acceso a recursos compartidos en la modalidad de exclusión mutua. En segundo lugar, deben poder formar grupos de procesos (los procesos independientes se consideran como grupos unitarios). En tercer lugar, los procesos no deben requerir sincronización (es decir, estar activos en sus respectivos procesadores al mismo tiempo) y deben tener estrictos requisitos de consenso para poder acceder a los recursos (se requiere un acuerdo para asignar consecutivamente los recursos solicitados por un proceso, es decir, una vez iniciada la secuencia de asignación de recursos, no se puede interrumpir para asignar recursos a otros procesos, hasta que el proceso activo libere los recursos).

Se trata de grupos de procesos distribuidos en nodos de proceso que acceden a recursos críticos. Estos recursos son compartidos en forma de exclusión mutua distribuida y debe decidirse, según la demanda de recursos de los procesos, cuáles serán las prioridades para asignar los recursos a los procesos que los requieran (para asignar en los procesos sólo se tendrán en cuenta los recursos disponibles, es decir, los que aún no se han asignado en determinados procesos): - El permiso de acceso a los recursos compartidos de un nodo no sólo dependerá de si los nodos los utilizan o no, sino también del valor de agregación de las preferencias (prioridades) de los distintos nodos en cuanto a la concesión de acceso a los recursos compartidos (alternativas). - Las opiniones (prioridades) de los distintos nodos en cuanto a la concesión de acceso a los recursos compartidos (alternativas) dependerán de la consideración del valor de las variables que representen el estado de cada uno de los distintos nodos. Cada nodo debe expresar sus prioridades para la asignación de los diferentes recursos compartidos de acuerdo con las necesidades de recursos de cada proceso (que pueden formar parte de un grupo de procesos).

Estos recursos compartidos disponibles alojados en diferentes nodos del sistema distribuido pueden ser requeridos por los procesos (agrupados o independientes) que se ejecutan en los nodos: Estado de cada proceso (independiente o perteneciente a un grupo de procesos). Estado de cada uno de los nodos (número de procesos, prioridades de los procesos, uso de la CPU, uso de la memoria principal, uso de la memoria virtual, memoria adicional requerida para cada recurso solicitado por cada proceso). Carga adicional estimada del procesador requerida para cada recurso solicitado por cada proceso (dependiendo de la disponibilidad de los datos). Carga adicional estimada de entrada/salida requerida para cada recurso solicitado por cada proceso (dependiendo de la disponibilidad de los datos). Estado de cada una de las acciones en el modo de exclusión mutua distribuida en el nodo (asignada a un proceso local o remoto, o disponible). Predisposición (prioridad



nodal) para conceder acceso a cada uno de los recursos compartidos  $r$  en la modalidad de exclusión mutua distribuida (resultará de la consideración de las variables representativas del estado del nodo, la prioridad de los procesos y la carga computacional adicional que supondría asignar el recurso al proceso solicitante). Carga actual del nodo, que puede calcularse como el promedio de los porcentajes de uso de la CPU, la memoria y las entradas/salidas en un momento dado (estos indicadores de carga pueden variar según el caso, algunos pueden añadirse o cambiarse); también deben definirse las categorías de carga actual, por ejemplo, Alta, Media y Baja, indicándose los rangos de valores de cada categoría.

En La Red Martínez (2017) se desarrolla un operador de agregación para la asignación de recursos en sistemas distribuidos, las premisas, estructuras de datos y el operador mencionado en esa publicación son aplicadas para resolver el primer escenario, que se describe a continuación.

Todas las figuras y tablas de este trabajo se basan en La Red Martínez (2017), Agostini, La Red Martínez y Acosta (2018) y Agostini, La Red Martínez y Acosta (2019).

### 3. Descripción del operador de agregación

El operador propuesto consta de las siguientes etapas:

1. Expresión de los valores calculados en términos de 2-tupla utilizando un conjunto de etiquetas lingüísticas.
2. Cálculo de la carga computacional actual de los nodos.
3. Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas.
4. Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo (se las calcula en cada nodo para cada proceso).
5. Cálculo de las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles (se las calcula en el administrador centralizado de recursos compartidos) y determinación del orden en que se asignarán los recursos y a qué proceso será asignado cada recurso.
6. Cálculo de las prioridades o preferencias de los procesos para acceder secuencialmente, a todos sus recursos compartidos.

#### 3.1. Expresión de los valores calculados en términos de 2-tupla usando un conjunto de etiquetas lingüísticas

El conjunto de etiquetas utilizado puede variar en cada nodo, es decir, que el conjunto es independiente del nodo central, y es este el encargado de realizar la traslación simbólica, según se indica en Dutta, Labella, Rodríguez y Martínez (2019), proceso que consiste en convertir los conjuntos de etiquetas recibidos en el nodo central, en el conjunto que este último trabaja, el cual debe ser el de mayor número de etiquetas. De igual manera, esa traslación simbólica se realiza cuando el nodo central devuelve la información final de prioridades a cada nodo, como se puede ver en la Figura 1.



Figura 1. Proceso de traslación de etiquetas que recibe y envía el nodo central. Fuente: Elaboración propia.





Las valoraciones expresadas en un formato lingüístico utilizando las etiquetas lingüísticas y semánticas mencionadas se pueden ver en Figura 2.



Figura 2. Representación del conjunto de etiquetas. Fuente: Elaboración propia.

En cada nodo se define una interfaz entre las aplicaciones y el sistema operativo, que a través de un Runtime (software en tiempo de ejecución complementario al sistema operativo) incluido en esa interfaz, gestiona los procesos y recursos compartidos y define el escenario correspondiente. Además, los Runtime interactúan entre sí para intercambiar información y en uno de los nodos hay un coordinador global de Runtime que evalúa y ejecuta el modelo de decisión y el operador de agregación correspondiente. A continuación, se describe cada uno de los pasos anteriores.

### 3.2. Cálculo de la carga actual de los nodos

Para obtener un indicador de la carga computacional actual de cada nodo, se pueden adoptar diferentes criterios; en esta propuesta los criterios serán el porcentaje de uso de la CPU, el porcentaje de uso de la memoria y el porcentaje de uso de la operación de entrada/salida. La carga computacional de cada nodo, el número de criterios para determinar la carga de los nodos, los criterios que se aplican y el cálculo de la carga computacional de cada nodo, son los mencionados en La Red Martínez (2017).

### 3.3. El establecimiento de las categorías de carga de cálculo y de los vectores de pesos asociados a ellos

Las categorías de carga computacional actuales de cada nodo, el número de categorías para determinar la carga de los nodos, las categorías que se aplican, los vectores de pesos asociados a las categorías de carga computacional actuales de cada nodo. En esta propuesta, los criterios serán los utilizados en La Red Martínez (2017).

Establecimiento de vectores de pesos (iguales para todos los nodos): pesos =  $\{w_{ij}\}$  con  $i = 1, \dots, a$  (número de categorías de carga computacional) y  $j = 1, \dots, e$  (número máximo de criterios).

### 3.4. Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo (se calculan en cada nodo para cada proceso y podrían denominarse prioridades nodales)

Estas prioridades se calculan en cada nodo para cada solicitud de recursos originada en cada proceso; el cálculo considera el vector de peso correspondiente según la carga actual del nodo y el vector de los valores concedidos por el nodo según los criterios de evaluación de la solicitud.

Los vectores de valoración que se aplicarán para cada solicitud de un recurso por un proceso, de acuerdo con los criterios establecidos para la determinación de la prioridad que en cada caso y momento fijará el nodo en el que se produce la solicitud, son los siguientes: valoraciones  $(r_{ij} p_{kl}) = \{cp_m\}$  con  $i = 1, \dots, n$  (nodo donde reside el recurso),  $j = 1, \dots, r$  (recurso en el nodo  $i$ ),  $k = 1, \dots, n$  (nodo donde reside el proceso),  $l = 1, \dots, p$  (proceso en el nodo  $k$ ) y  $m = 1, \dots, e$  (criterio de valoración de la prioridad de la solicitud).

Estos valores son representados en el formato de 2-tupla, considerando las etiquetas lingüísticas propuestas anteriormente. Por lo tanto, el valor de cada criterio deberá compararse con el valor medio de cada etiqueta, la diferencia mínima de esa comparación será la etiqueta apropiada.

El primer elemento de la 2-tupla será el valor lingüístico de esa etiqueta. El segundo elemento será la



diferencia entre el valor de los criterios buscados y el valor medio de la etiqueta seleccionada.

$dm$  = la diferencia mínima entre las diferencias de  $cp_m$  y el valor más representativo de cada etiqueta lingüística.

Recursos - Procesos		Criterio 2-tupla			
$r_1; p_{11}$	T(etiqueta <sub>n</sub> ; dm <sub>n</sub> )	...	T(etiqueta <sub>m</sub> ; dm <sub>m</sub> )	...	T(etiqueta <sub>n</sub> ; dm <sub>n</sub> )
...	...	...	...	...	...
$r_j; p_{j1}$	T(etiqueta <sub>n</sub> ; dm <sub>n</sub> )	...	T(etiqueta <sub>m</sub> ; dm <sub>m</sub> )	...	T(etiqueta <sub>n</sub> ; dm <sub>n</sub> )
...	...	...	...	...	...
$r_n; p_{n1}$	T(etiqueta <sub>n</sub> ; dm <sub>n</sub> )	...	T(etiqueta <sub>m</sub> ; dm <sub>m</sub> )	...	T(etiqueta <sub>n</sub> ; dm <sub>n</sub> )

Tabla 1. Valoraciones asignadas a los criterios de cálculo de la prioridad o preferencia que cada nodo dará a cada requerimiento de cada proceso según la carga del nodo en 2-tuplas. Fuente: Elaboración propia.

Valoraciones de la etiqueta  $(r_{ij}; p_{kl}) = 2\text{-tupla} = T(\text{etiqueta}_m; dm_m)$  donde el subíndice  $m$  corresponde a las etiquetas lingüísticas definidas anteriormente, como puede verse en la Tabla 1.

En resumen, la prioridad nodal (que debe calcularse en el nodo en el que se produce la solicitud) de un proceso de acceso a un recurso determinado (que puede ser en cualquier nodo) se calcula mediante el producto escalar de los vectores mencionados: prioridad nodal  $(r_{ij}; p_{kl}) = \sum w_{om} * T(\text{etiqueta}_m; dm_m) = T(\text{etiqueta}_n; dm_n) = TPN_{ijkl}$  (Tupla de Prioridad Nodal) con la  $o$  indicando el vector de pesos según la carga del nodo, manteniendo todos los demás subíndices los significados explicados anteriormente. Con  $m$  y  $n$  indicando la etiqueta lingüística correspondiente dentro del conjunto adoptado definido anteriormente.

Esta prioridad nodal debe transformarse en el formato de 2-tuplas, teniendo en cuenta las etiquetas lingüísticas ya mencionadas. Por lo tanto, será necesario comparar cada valor de prioridad nodal con el valor medio de cada etiqueta, la diferencia mínima de estas comparaciones indicará la etiqueta correspondiente.

### 3.5. Cálculo de las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles (se las calcula en el administrador centralizado de recursos compartidos) y determinación del orden en que se asignarán los recursos y a qué proceso será asignado cada recurso

La Tabla 2 se utiliza para calcular las prioridades finales, en el que se colocan las prioridades o preferencias nodales calculadas en la etapa anterior; en esta tabla cada fila contiene la información de las prioridades nodales de los diferentes procesos para acceder a un determinado recurso.

Recurso s	2-tupla		
$r_1$	TPN <sub>111</sub>	...	TPN <sub>112</sub> ... TPN <sub>113</sub>
...	...	...	...
$r_j$	TPN <sub>j11</sub>	...	TPN <sub>j12</sub> ... TPN <sub>j13</sub>
...	...	...	...
$r_n$	TPN <sub>n11</sub>	...	TPN <sub>n12</sub> ... TPN <sub>n13</sub>

Tabla 2. Prioridades nodales de los procesos para acceder a cada recurso en 2-tupla. Fuente: Elaboración propia.

A continuación, es necesario calcular el vector de pesos finales que se utilizará en el proceso de agregación para determinar el orden o la prioridad de acceso a los recursos.

Pesos finales =  $\{wf_{kl}\}$  con  $k = 1, \dots, n$  (número de nodos) y  $l = 1, \dots, p$  (número máximo de procesos por nodo), donde  $np$  es el número de procesos en el sistema y  $pr_{gl}$  es la prioridad del grupo de procesos al que pertenece el proceso.





El siguiente paso es normalizar los pesos recién obtenidos dividiendo cada uno de ellos por la suma de todos.

Así se obtiene un vector de peso normalizado (en el rango de 0 a 1 inclusive) y con la restricción de que la suma de los elementos del vector debe dar 1:

$$\sum \{nwf_{kl}\} = 1 \text{ con } k = 1, \dots, n \text{ (número de nodos) y } l = 1, \dots, p \text{ (número máximo de procesos por nodo).}$$

Las prioridades nodales tomadas fila por fila para cada recurso serán multiplicadas escalarmente por el vector de peso final normalizado. De esta manera es posible obtener las prioridades global final de acceso de cada proceso a cada recurso. A continuación se indica cómo se obtiene el orden o la prioridad con que se asignarán los recursos y a qué proceso se asignará cada uno.

Prioridad global final  $(r_{ij}, p_{kl}) = TPN_{ijkl} = TPGF_{ijkl}$  (Tupla de prioridad global final) con el  $r_{ij}$  indicando el recurso  $j$  del nodo  $i$ ,  $TPN_{ijkl}$  es el formato de 2-tuplas,  $ij$  indicando el recurso  $j$  del nodo  $i$ ,  $kl$  el proceso  $l$  del nodo  $k$  y el producto de la prioridad global final del proceso para acceder a dicho recurso, como se puede ver en la Tabla 3.

Recurso $i$	Prioridades Nodales de los Procesos				
$r_{i1}$	TPGF <sub>111}</sub>	...	TPGF <sub>11k}</sub>	...	TPGF <sub>11p}</sub>
...	...	...	...	...	...
$r_{ij}$	TPGF <sub>j11}</sub>	...	TPGF <sub>jkl}</sub>	...	TPGF <sub>jpp}</sub>
...	...	...	...	...	...
$r_{in}$	TPGF <sub>n11}</sub>	...	TPGF <sub>nkl}</sub>	...	TPGF <sub>npp}</sub>

Tabla 3. Tupla de Prioridad Global Final. Fuente: Elaboración propia.

El siguiente paso es normalizar la Tabla 3 entre los valores extremos. Esto se hará utilizando los valores máximo, mínimo y rango calculados de la Tabla 3 y representados en la Tabla 4.

Etiqueta	Valor
Valor Máximo	Máximo (TPGF <sub>jkl</sub> )
Valor mínimo	Mínimo (TPGF <sub>jkl</sub> )
Rango	Máximo (TPGF <sub>jkl</sub> ) - Mínimo (TPGF <sub>jkl</sub> )

Tabla 4. Cálculo de los valores máximo, mínimo y el rango. Fuente: Elaboración propia.

Para normalizar, el valor numérico de cada 2-tupla debe restarse del valor mínimo y dividirse por el rango, que es la diferencia entre el valor máximo y el valor mínimo, para obtener el TPGFN (Tupla de Prioridad Global Final Normalizada), esto puede verse en el Tabla 5.

Recursos	2-tupla				
$r_{i1}$	TPGFN <sub>111}</sub>	...	TPGFN <sub>11k}</sub>	...	TPGFN <sub>11p}</sub>
...	...	...	...	...	...
$r_{ij}$	TPGFN <sub>j11}</sub>	...	TPGFN <sub>jkl}</sub>	...	TPGFN <sub>jpp}</sub>
...	...	...	...	...	...
$r_{in}$	TPGFN <sub>n11}</sub>	...	TPGFN <sub>nkl}</sub>	...	TPGFN <sub>npp}</sub>

Tabla 5. Tupla de Prioridad Global Final Normalizada. Fuente: Elaboración propia.

El mayor de estos productos realizados para los diferentes procesos en relación con el mismo recurso indicará cuál de los procesos tendrá acceso al recurso.

La suma de todos estos productos en relación con el mismo recurso indicará la prioridad que se le asignará a ese recurso, en relación con los demás recursos que también deberán asignarse. Esto es lo que se denominará Función de Asignación de Sistemas Distribuidos Lingüística (FASDL):



$$\text{FASDL}(r_{ij}) = \sum \text{TPGFN}_{ijkl} = \text{prioridad de asignación de recursos } r_{ij}$$

Calculando el FASDL para todos los recursos se obtendrá un vector 2-tupla, y ordenando sus elementos de mayor a menor, se obtendrá el orden de prioridad de asignación de recursos. Estos deben ser normalizados garantizando que las 2-tuplas obtenidas se encuentran en el intervalo [0, 1], para ello, se utilizarán los valores máximo, mínimo y rango, como puede verse en la Tabla 6.

Etiqueta	Valor
Valor Máximo	Máximo (FASDL <sub>μ</sub> )
Valor mínimo	Mínimo (FASDL <sub>μ</sub> )
Rango	Máximo (FASDL <sub>μ</sub> ) – Mínimo (FASDL <sub>μ</sub> )

Tabla 6. Valoraciones para normalizar el FASDL. Fuente: Elaboración propia.

Además, como ya se ha indicado, el mayor de los productos TPGFN<sub>ijkl</sub> para cada recurso indicará el proceso al que se asignará el recurso. Como se puede observar en la Figura 3.

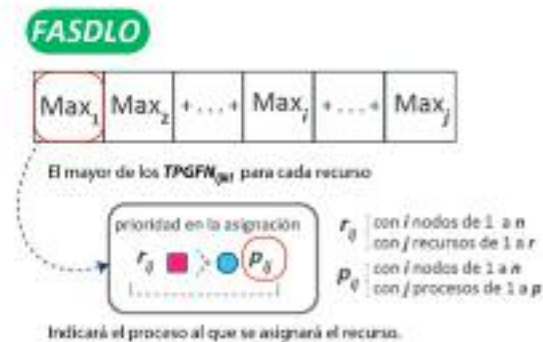


Figura 3. Orden de prioridad de cada proceso sobre cada recurso. Fuente: Elaboración propia.

Esto es lo que se denominará Función de Asignación de Sistemas Distribuidos Lingüísticos Normalizados (FASDLN):

$\text{FASDLN}(r_{ij}) = \sum (\text{TPGFN}_{ijkl} / (\text{máximo}(\text{TPGFN}_{ijkl}) - \text{mínimo}(\text{TPGFN}_{ijkl}))) = \text{prioridad de asignación de recursos } r_{ij} \text{ normalizada entre valores extremos, como puede verse en la Tabla 7.}$

Orden de asignación de los recursos	Proceso al que se asignará el recurso
1º: $r_{ij}$ del $\text{Max}(\text{FASDLN}(r_{ij}))$	$p_{ij}$ del $\text{Max}(\text{TPGFN}_{μij})$ para el $r_{ij}$ seleccionado
2º: $r_{ij}$ del $\text{Max}(\text{FASDLN}(r_{ij}))$	$p_{ij}$ del $\text{Max}(\text{TPGFN}_{μij})$ para el $r_{ij}$ seleccionado
para los $r_{ij}$ no asignados	
....	....
último: $r_{ij}$ no asignado	$p_{ij}$ del $\text{Max}(\text{TPGFN}_{μij})$ para el $r_{ij}$ seleccionado

Tabla 7. Orden o prioridad final de asignación de recursos y proceso al que se asigna cada recurso en la primera iteración. Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas; cabe señalar que los recursos asignados estarán disponibles una vez que los procesos los liberen y, por lo tanto, podrán asignarse a otros procesos. Se debe recalcular la Tabla 6 omitiendo las asignaciones de recursos ya hechas.

### 3.6. Cálculo de las prioridades o preferencias de los procesos para acceder secuencialmente, a todos sus recursos compartidos

El primer paso es obtener las prioridades globales definitivas para asignar los recursos (Función de Asignación para Sistemas Distribuidos Lingüística Normalizada - FASDLN). Seguidamente establecer el orden o prioridad de asignación de los recursos y el proceso al que se asigna cada recurso (Función de Asignación



para Sistemas Distribuidos Lingüística Normalizada Ordenada - FASDLNO) puede verse en la Tabla 7. La unión de todas las asignaciones ordenadas, realizadas en todas las rondas (FASDLNO) conforman lo que se denomina (Función de Asignación para Sistemas Distribuidos Lingüística Normalizada Concatenada - FASDLNC)), como se muestra en la Figura 4.



Figura 4. Pasos para obtener la FASDLN, FASDLNO y FASDLNC. Fuente: Elaboración propia.

La tabla FASDLNC, se obtendrá a partir de la concatenación de las tablas FASDLNO de cada iteración, según se muestra en la Tabla 8.

FASDLNO	Iteraciones
FASDLNO	Filas de 1 a n
1er iteración	n = número de filas de la FASDLNO en la primer iteración
FASDLNO	Filas de n+1 a m
2da iteración	m = número de filas de la FASDLNO en la segunda iteración
...	...
FASDLNO	Filas de m+1 a t
última iteración	t = número de filas de la FASDLNO en la última iteración

Tabla 8. Concatenación de las tablas de asignación ordenada (FASDLNC). Fuente: Elaboración propia.

**Prioridad global final del proceso**

Una vez completada la tabla FASDLNC (Tabla 8), se calculará las prioridades globales finales de los procesos para acceder a sus recursos y se establecerá el orden en el que se ejecutará cada uno, recibiendo todos los recursos solicitados.



Figura 5. Pasos para obtener desde la FASDLNC a la FASDLNCO. Fuente: Elaboración propia.

En la Figura 5 se muestra la serie de pasos que se debe realizar, se tendrá en cuenta la tabla FASDLNC, con todos los procesos y recursos vinculados. Se sumará la prioridad de todas las asignaciones recurso/proceso, para cada proceso, y se las dividirá por la cantidad de asignaciones del mismo. El proceso que tenga mayor prioridad global final recibirá primero los recursos solicitados. Esto constituye lo que se denominará Prioridad Global Final Lingüística del Proceso (PGFLP), como se muestra en Figura 6:



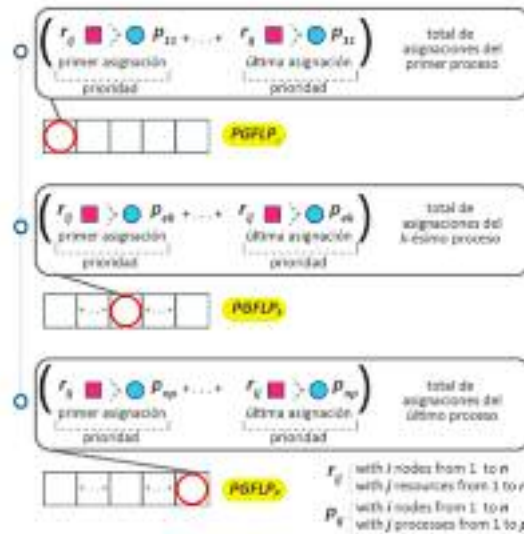


Figura 6. Cálculo de la PGFLP de cada proceso. Fuente: Elaboración propia.

$PGFLP_i = i = 1, \dots, h = \frac{\sum_{j=1}^n r_{ij} p_{ij}}{\text{Cantidad de conexiones global del proceso } i}$   $i = \text{total de procesos en el sistema (sumatoria de procesos de los nodos), } j = n^{\circ} \text{ de recursos asignados al proceso } i.$   
 Del vector **PGFLP** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario global de asignación de recursos a procesos, como se muestra en Figura 7.  
**PGFLPO** = Prioridad Global Final Lingüística del Proceso Ordenada  
 $j = \text{cardinalidad de PGFLP (total de procesos en el sistema)}$   
 $PGFLPO_i = \text{Max (PGFLP no ordenado)} \quad i = 1, \dots, j$   
 No ordenado =  $PGFLP_i \neq PGFLPO$   
 1ero:  $PGFLPO_1 = \text{Max (PGFLP)} \quad i = 1, \dots, j$   
 2do:  $PGFLPO_2 = \text{Max (no ordenado PGFLP)} \quad i = 1, \dots, j$   
 ...  
 último:  $PGFLPO_j = \text{Max (no ordenado PGFLP)} \quad i = 1, \dots, j$

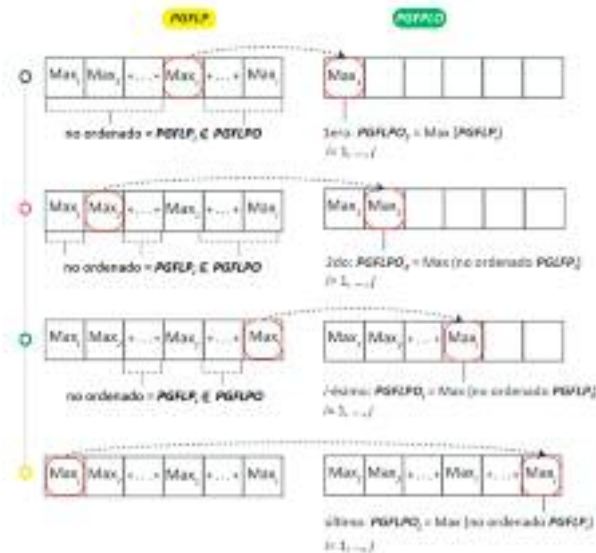


Figura 7. Ejemplo para calcular la PGFLPO de cada proceso. Fuente: Elaboración propia.



#### Función de Asignación para Sistemas Distribuidos Lingüísticos Normalizados Concéntricos Ordenados (FASDLNCO)

La FASDLNCO establecerá el orden de asignación por prioridades globales finales de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno, recibiendo todos los recursos solicitados. Para ello, se tendrá en cuenta la tabla FASDLNC y la PGFLPO.

Se calcularán las cardinalidades (número de asignaciones de recursos a cada proceso) obtenidas de cada uno de los procesos del vector PGFLPO en la tabla FASDLNC.

$cp_i$ = cardinalidad proceso(PGFLPO<sub>i</sub>) en FASDLNC

Luego, habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla FASDLNC de cada uno de los procesos del vector PGFLPO. El total de elementos por cada proceso, estará determinado por la cardinalidad calculada en el paso anterior, como se puede ver en Fig. 8.

FASDLNCO<sub>1</sub> formado por el primer elemento de la FASDLNC para el proceso(PGFLPO<sub>1</sub>) y FASDLNCO<sub>1, cp<sub>1</sub></sub> el último elemento de la FASDLNC para el proceso(PGFLPO<sub>1</sub>).

FASDLNCO<sub>2, cp<sub>2</sub></sub> formado por el primer elemento de la FASDLNC para el proceso(PGFLPO<sub>2</sub>) y FASDLNCO<sub>2, cp<sub>2</sub></sub> el último elemento de la FASDLNC para el proceso(PGFLPO<sub>2</sub>).

Se continúa el cálculo para todos los procesos intermedios.

FASDLNCO<sub>cp<sub>1</sub>+cp<sub>2</sub>+...+cp<sub>i-1</sub></sub> formado por el primer elemento de la FASDLNC para el proceso(PGFLPO<sub>i</sub>) y FASDLNCO<sub>cp<sub>1</sub>+cp<sub>2</sub>+...+cp<sub>i</sub></sub> el último elemento de la FASDLNC para el proceso(PGFLPO<sub>i</sub>).

FASDLNCO<sub>cp<sub>1</sub>+cp<sub>2</sub>+...+cp<sub>i-1</sub></sub> formado por el primer elemento de la FASDLNC para el proceso(PGFLPO<sub>i</sub>) y FASDLNCO<sub>cp<sub>1</sub>+cp<sub>2</sub>+...+cp<sub>i</sub></sub> el último elemento de la FASDLNC para el proceso(PGFLPO<sub>i</sub>).

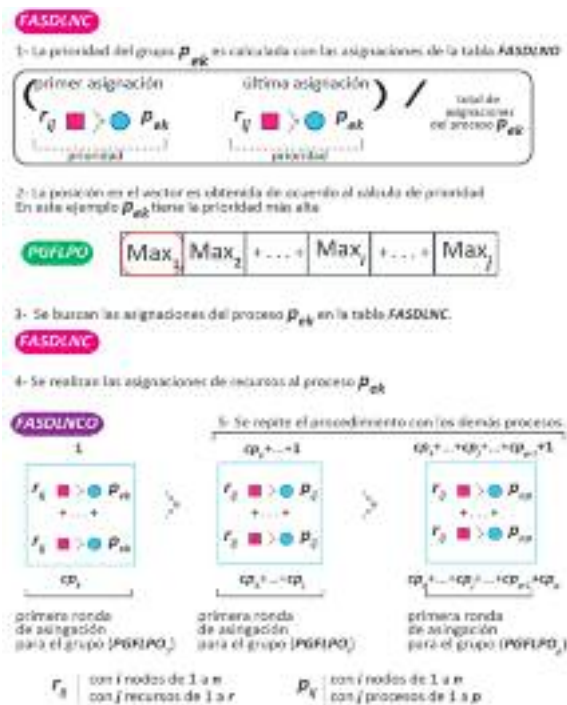


Figura 8. Esquema para obtener la FASDLNCO de cada proceso. Fuente: Elaboración propia.

De esta manera, se ha dado respuesta a todas las solicitudes de recursos de todos los procesos, respetando la exclusión mutua y las prioridades de los procesos, las prioridades nodales y las prioridades finales, teniendo en cuenta los estrictos requisitos de consenso establecidos para este escenario.

## 4. Consideraciones para las operaciones de agregación

Las características de las operaciones de agregación descritas permiten considerar que el método propuesto pertenece a la familia de operadores de agregación Neat-OWA, que se caracterizan por Yager (1988) y Yager y Pasi (2002).

## 5. Ejemplo y discusión de los resultados

En esta sección se explicará en detalle un ejemplo de aplicación del operador de agregación propuesto. El











En resumen, la prioridad nodal (que debe calcularse en el nodo en que se produce la solicitud) de un proceso para acceder a un recurso determinado (que puede estar en cualquier nodo) se calcula mediante el producto escalar de los vectores mencionados anteriormente.

### 5.3. Cálculo de las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles (calculados en el gestor centralizado de recursos) y determinación del orden de asignación de los recursos y del proceso de asignación de cada uno de ellos

La Tabla 14 y Tabla 15 y Tabla 16 se utiliza para calcular las prioridades finales, en el que se colocan las prioridades o preferencias nodales calculadas en la etapa anterior; en esta tabla cada fila contiene la información de las prioridades nodales de los diferentes procesos para acceder a un determinado recurso.

Recursos	Prioridades Nodales de los Procesos				
	P <sub>11</sub>	P <sub>12</sub>	P <sub>13</sub>	P <sub>14</sub>	P <sub>15</sub>
R <sub>11</sub>	TPN(A,0.0483)	TPN(MA,-0.0083)	TPN(A,0.0483)	-	-
R <sub>12</sub>	TPN(M,-0.005)	TPN(A,-0.0467)	TPN(M,0.035)	TPN(M,-0.0825)	-
R <sub>13</sub>	-	-	TPN(A,0.0733)	TPN(A,-0.0592)	-
R <sub>14</sub>	TPN(B,0.0217)	TPN(A,-0.0267)	TPN(M,-0.06)	-	TPN(A,-0.0892)
R <sub>15</sub>	TPN(M,-0.015)	TPN(A,-0.0267)	TPN(M,-0.05)	TPN(A,-0.0617)	TPN(A,0.0308)
R <sub>21</sub>	TPN(MA,-0.0483)	-	-	TPN(M,0.0725)	-
R <sub>22</sub>	TPN(B,0.0717)	-	-	-	-
R <sub>23</sub>	-	TPN(M,0.06)	TPN(A,-0.0367)	TPN(A,0.0483)	TPN(M,-0.0525)
R <sub>24</sub>	-	-	TPN(A,-0.0667)	-	-
R <sub>25</sub>	-	TPN(A,0.0333)	TPN(A,-0.0517)	TPN(A,-0.0167)	TPN(M,0.0075)

Tabla 14. Prioridades nodales de los procesos para acceder a cada recurso en 24hrs (m, ma, pa, pm, pv). Fuente: Elaboración propia.

Recursos	Prioridades Nodales de los Procesos				
	P <sub>21</sub>	P <sub>22</sub>	P <sub>23</sub>	P <sub>24</sub>	P <sub>25</sub>
R <sub>11</sub>	-	TPN(A,-0.0367)	-	-	TPN(A,0.0733)
R <sub>12</sub>	TPN(A,-0.0667)	TPN(A,0.0258)	-	-	TPN(A,-0.0142)
R <sub>13</sub>	-	-	-	TPN(MA,-0.0808)	TPN(MA,0.0117)
R <sub>14</sub>	-	-	TPN(A,-0.0167)	-	-
R <sub>15</sub>	-	TPN(A,-0.0292)	-	-	TPN(MA,0.0017)
R <sub>21</sub>	TPN(B,-0.0308)	TPN(A,-0.0392)	-	-	-
R <sub>22</sub>	TPN(M,-0.04)	-	-	TPN(A,-0.0817)	-
R <sub>23</sub>	TPN(M,0.02)	-	-	-	-
R <sub>24</sub>	TPN(A,-0.0317)	-	-	TPN(A,-0.0117)	-

Tabla 15. Prioridades nodales de los procesos para acceder a cada recurso en 24hrs (m, ma, pa, pm, pv). Fuente: Elaboración propia.

Recursos	Prioridades Nodales de los Procesos				
	P <sub>31</sub>	P <sub>32</sub>	P <sub>33</sub>	P <sub>34</sub>	P <sub>35</sub>
R <sub>11</sub>	TPN(A,0.0258)	-	-	TPN(A,0.0583)	TPN(A,0.0533)
R <sub>12</sub>	TPN(MA,-0.0483)	TPN(A,0.0058)	TPN(M,0.0025)	TPN(A,0.0808)	TPN(MA,-0.0733)
R <sub>13</sub>	TPN(A,-0.0142)	TPN(A,0.0558)	TPN(A,0.0308)	TPN(MA,-0.0183)	-
R <sub>14</sub>	TPN(A,0.0633)	-	-	TPN(A,-0.0017)	TPN(A,0.0283)
R <sub>15</sub>	TPN(A,-0.0162)	TPN(MA,-0.0708)	TPN(A,-0.0042)	TPN(B,0.0442)	-
R <sub>21</sub>	TPN(A,0.0808)	TPN(A,0.0083)	-	-	-
R <sub>22</sub>	-	TPN(M,0.07)	TPN(A,0.0108)	TPN(A,-0.0182)	-
R <sub>23</sub>	-	TPN(A,0.0258)	TPN(A,0.0208)	TPN(MA,-0.0383)	-
R <sub>24</sub>	TPN(A,-0.0142)	TPN(A,0.0083)	TPN(A,0.0208)	TPN(A,0.0658)	TPN(A,-0.0192)
R <sub>25</sub>	TPN(A,0.0583)	TPN(A,-0.0492)	TPN(A,0.0433)	TPN(A,-0.0592)	TPN(A,-0.0542)

Tabla 16. Prioridades nodales de los procesos para acceder a cada recurso en 24hrs (m, ma, pa, pm, pv). Fuente: Elaboración propia.

A continuación, se debe calcular el vector de peso final que se utilizará en el proceso de agregación final para determinar el orden o la prioridad de acceso a los recursos. Además, los pesos recientemente obtenidos deberán normalizarse dividiendo cada uno de ellos por la suma de todos ellos.



	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$
$\beta_1$	0.2	0.1300	0.2	0.1303	0.1303
$\beta_2$	0.0950	0.0945	0.2988	0.0645	0.0645

Tabla 18. Vector de pesos ( $\beta_1, \beta_2, \beta_3, \beta_4, \beta_5$ )

	$\beta_6$	$\beta_7$	$\beta_8$	$\beta_9$	$\beta_{10}$
$\beta_6$	0.2	0.0907	0.2	0.1303	0.0667
$\beta_7$	0.0950	0.0323	0.2988	0.0645	0.0323

Tabla 19. Vector de pesos ( $\beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}$ )

	$\beta_{11}$	$\beta_{12}$	$\beta_{13}$	$\beta_{14}$	$\beta_{15}$	Total
$\beta_{11}$	0.0057	0.2	0.0657	0.0657	0.2	2.0657
$\beta_{12}$	0.0323	0.0988	0.0323	0.0323	0.0688	1

Tabla 17. Vector de pesos ( $\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}$ ). Fuente: Elaboración propia.

Las prioridades nodales indicadas en la Tabla 14, Tabla 15 y Tabla 16 tomadas fila por fila, es decir, para cada recurso, se multiplicarán por el vector de peso normalizado (nwpij) de la Tabla 17, Tabla 18 y Tabla 19. Esto se puede ver en la Tabla 20, Tabla 21 y Tabla 22.

Recurso	Prioridades Nodales de los Procesos				
	$\beta_{11}$	$\beta_{12}$	$\beta_{13}$	$\beta_{14}$	$\beta_{15}$
$P_{11}$	TPN(ED:0.089)	TPN(ED:0.053)	TPN(ED:0.089)	-	-
$P_{12}$	TPN(ED:0.046)	TPN(ED:0.046)	TPN(ED:0.052)	TPN(ED:0.027)	-
$P_{13}$	-	-	TPN(ED:0.072)	TPN(ED:0.038)	-
$P_{14}$	TPN(ED:0.034)	TPN(ED:0.041)	TPN(ED:0.042)	-	TPN(ED:0.038)
$P_{15}$	TPN(ED:0.047)	TPN(ED:0.047)	TPN(ED:0.044)	TPN(ED:0.038)	TPN(ED:0.048)
$P_{16}$	TPN(ED:0.076)	-	-	TPN(ED:0.037)	-
$P_{17}$	TPN(ED:0.039)	-	-	-	-
$P_{18}$	-	TPN(ED:0.037)	TPN(ED:0.051)	TPN(ED:0.046)	TPN(ED:0.038)
$P_{19}$	-	-	TPN(ED:0.038)	-	-
$P_{20}$	-	TPN(ED:0.045)	TPN(ED:0.06)	TPN(ED:0.046)	TPN(ED:0.033)

Tabla 20. Prioridades nodales de los procesos para acceder a cada recurso en 2-tupla, multiplicado por su vector de peso ( $\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}$ ). Fuente: Elaboración propia.

Recurso	Prioridades Nodales de los Procesos				
	$\beta_{11}$	$\beta_{12}$	$\beta_{13}$	$\beta_{14}$	$\beta_{15}$
$P_{21}$	-	TPN(ED:0.020)	-	-	TPN(ED:0.024)
$P_{22}$	TPN(ED:0.058)	TPN(ED:0.022)	-	-	TPN(ED:0.021)
$P_{23}$	-	-	-	TPN(ED:0.05)	TPN(ED:0.027)
$P_{24}$	-	-	TPN(ED:0.02)	-	-
$P_{25}$	-	-	-	-	-
$P_{26}$	-	TPN(ED:0.021)	-	-	TPN(ED:0.027)
$P_{27}$	TPN(ED:0.032)	TPN(ED:0.020)	-	-	-
$P_{28}$	TPN(ED:0.045)	-	-	TPN(ED:0.038)	-
$P_{29}$	TPN(ED:0.050)	-	-	-	-
$P_{30}$	TPN(ED:0.067)	-	-	TPN(ED:0.042)	-

Tabla 21. Prioridades nodales de los procesos para acceder a cada recurso en 2-tupla, multiplicado por su vector de peso ( $\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}$ ). Fuente: Elaboración propia.

Recurso	Prioridades Nodales de los Procesos				
	$\beta_{11}$	$\beta_{12}$	$\beta_{13}$	$\beta_{14}$	$\beta_{15}$
$P_{31}$	TPN(ED:0.022)	-	-	TPN(ED:0.025)	TPN(ED:0.075)
$P_{32}$	TPN(ED:0.020)	TPN(ED:0.068)	TPN(ED:0.017)	TPN(ED:0.024)	TPN(ED:0.074)
$P_{33}$	TPN(ED:0.021)	TPN(ED:0.071)	TPN(ED:0.020)	TPN(ED:0.026)	-
$P_{34}$	TPN(ED:0.022)	-	-	TPN(ED:0.021)	TPN(ED:0.067)
$P_{35}$	TPN(ED:0.021)	TPN(ED:0.074)	TPN(ED:0.021)	TPN(ED:0.022)	-
$P_{36}$	TPN(ED:0.023)	TPN(ED:0.068)	-	-	-
$P_{37}$	-	TPN(ED:0.055)	TPN(ED:0.022)	TPN(ED:0.021)	-
$P_{38}$	-	TPN(ED:0.067)	TPN(ED:0.022)	TPN(ED:0.026)	-
$P_{39}$	TPN(ED:0.021)	TPN(ED:0.065)	TPN(ED:0.022)	TPN(ED:0.024)	TPN(ED:0.063)
$P_{40}$	TPN(ED:0.022)	TPN(ED:0.06)	TPN(ED:0.022)	TPN(ED:0.02)	TPN(ED:0.058)

Tabla 22. Prioridades nodales de los procesos para acceder a cada recurso en 2-tupla, multiplicado por su vector de peso ( $\beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}$ ). Fuente: Elaboración propia.

El siguiente paso es normalizar la Tabla 20, Tabla 21 y Tabla 22 entre los valores extremos. Para ello, se debe restar el valor numérico de la 2-tupla por el valor mínimo de todas ellas y dividirlo por el rango, que es la diferencia entre el valor máximo y el valor mínimo de las tablas ya mencionadas. Esto se puede ver en la Tabla 23.

Etiqueta	Valor
Valor Máximo	0.0760
Valor Mínimo	0.0122
Rango	0.0638

Tabla 23. Valoraciones para normalizar el FASDL. Fuente: Elaboración propia.





El resultado de esta normalización se puede ver en la Tabla 24, Tabla 25 y Tabla 26.

Recursos	Prioridades Nodales de los Procesos				
	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
A <sub>1</sub>	TPGFN(MA,0.06)	TPGFN(A,0.023)	TPGFN(MA,0.06)	-	TPGFN(MA,0.06)
A <sub>2</sub>	TPGFN(M,0.06)	TPGFN(M,0.06)	TPGFN(A,0.04)	TPGFN(M,0.06)	TPGFN(M,0.06)
A <sub>3</sub>	-	-	TPGFN(EA,0.06)	TPGFN(M,0.07)	-
A <sub>4</sub>	TPGFN(B,0.014)	TPGFN(M,0.04)	TPGFN(M,0.02)	-	TPGFN(B,0.014)
A <sub>5</sub>	TPGFN(M,0.04)	TPGFN(M,0.04)	TPGFN(M,0.02)	TPGFN(M,0.07)	TPGFN(M,0.04)
A <sub>6</sub>	TPGFN(EA,0.0)	-	-	TPGFN(B,0.05)	TPGFN(EA,0.0)
A <sub>7</sub>	TPGFN(M,0.016)	-	-	-	TPGFN(M,0.016)
A <sub>8</sub>	-	TPGFN(B,0.02)	TPGFN(MA,0.06)	TPGFN(M,0.02)	-
A <sub>9</sub>	-	-	TPGFN(A,0.02)	-	-
A <sub>10</sub>	-	TPGFN(M,0.017)	TPGFN(A,0.015)	TPGFN(M,0.004)	-

Tabla 24. Prioridades nodales de los procesos para acceder a cada recurso en 2-tupla normalizada (P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>, P<sub>5</sub>). Fuente: Elaboración propia.

Recursos	Prioridades Nodales de los Procesos				
	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
A <sub>1</sub>	-	TPGFN(MB,0.038)	-	-	TPGFN(MB,0.011)
A <sub>2</sub>	TPGFN(MA,0.062)	TPGFN(MB,0.007)	-	-	TPGFN(MB,0.038)
A <sub>3</sub>	-	-	-	TPGFN(A,0.018)	TPGFN(MB,0.010)
A <sub>4</sub>	-	-	TPGFN(MA,0.038)	-	-
A <sub>5</sub>	-	TPGFN(MB,0.038)	-	-	TPGFN(MB,0.06)
A <sub>6</sub>	TPGFN(B,0.007)	TPGFN(MB,0.045)	-	-	-
A <sub>7</sub>	TPGFN(M,0.051)	-	-	TPGFN(B,0.007)	-
A <sub>8</sub>	TPGFN(A,0.011)	-	-	-	-
A <sub>9</sub>	TPGFN(MA,0.061)	-	-	TPGFN(M,0.028)	-

Tabla 25. Prioridades nodales de los procesos para acceder a cada recurso en 2-tupla normalizada (P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>, P<sub>5</sub>). Fuente: Elaboración propia.

Recursos	Prioridades Nodales de los Procesos				
	P <sub>1</sub>	P <sub>2</sub>	P <sub>3</sub>	P <sub>4</sub>	P <sub>5</sub>
A <sub>1</sub>	TPGFN(MB,0.007)	-	-	TPGFN(MB,0.008)	TPGFN(MA,0.06)
A <sub>2</sub>	TPGFN(MB,0.008)	TPGFN(MA,0.064)	TPGFN(B,0.075)	TPGFN(MB,0.02)	TPGFN(EA,0.038)
A <sub>3</sub>	TPGFN(MB,0.038)	TPGFN(A,0.042)	TPGFN(MB,0.008)	TPGFN(MB,0.08)	-
A <sub>4</sub>	TPGFN(MB,0.019)	-	-	TPGFN(MB,0.021)	TPGFN(MA,0.03)
A <sub>5</sub>	TPGFN(MB,0.033)	TPGFN(EA,0.034)	TPGFN(MB,0.022)	TPGFN(ED,0.00)	-
A <sub>6</sub>	TPGFN(MB,0.01)	TPGFN(MA,0.03)	-	-	-
A <sub>7</sub>	-	TPGFN(A,0.007)	TPGFN(MB,0.015)	TPGFN(MB,0.006)	-
A <sub>8</sub>	-	TPGFN(MA,0.03)	TPGFN(MB,0.01)	TPGFN(MB,0.04)	-
A <sub>9</sub>	TPGFN(MB,0.038)	TPGFN(MA,0.03)	TPGFN(MB,0.01)	TPGFN(MB,0.013)	TPGFN(MA,0.042)
A <sub>10</sub>	TPGFN(MB,0.008)	TPGFN(A,0.018)	TPGFN(MB,0.001)	TPGFN(MB,0.006)	TPGFN(A,0.07)

Tabla 26. Prioridades nodales de los procesos para acceder a cada recurso en 2-tupla normalizada (P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>, P<sub>4</sub>, P<sub>5</sub>). Fuente: Elaboración propia.

El mayor de estos productos realizados para los diferentes procesos en relación con el mismo recurso indicará cuál de los procesos tendrá acceso al recurso.

La suma de todos estos productos en relación con el mismo recurso indicará la prioridad que deberá asignarse a este recurso, en relación con los demás recursos que también deberán asignarse. Esto constituye la Función de Asignación del Sistema Distribuido Lingüística (FASDL):

$$FASDL(r_{ij}) = \sum TPGFN_{ijkl} = r_{ij} \text{ prioridad de asignación de recursos.}$$

Al calcular la FASDL para todos los recursos, se obtendrá un vector de 2-tuplas y, ordenando sus elementos de mayor a menor, se obtendrá el orden de prioridad de asignación de recursos, que deberá ser normalizado asegurando que las 2-tuplas obtenidas estén en el intervalo [0, 1]. Esto se puede observar en la Tabla 27.

Etiqueta	Valor
Valor Máximo	5.2364
Valor Mínimo	1.8255
Rango	3.4109

Tabla 27. Valoraciones para normalizar el FASDL. Fuente: Elaboración propia.

Además, como se ha indicado anteriormente, el mayor de los TPGFN<sub>ijkl</sub> de cada recurso indicará el proceso al que se asignará el recurso.

El resultado de la normalización de las 2-tuplas constituye lo que se denominará Función de Asignación



del Sistema Distribuido Lingüística Normalizada a (FASDLN):

$FASDLN(r_{ij}) = \Sigma (TPGFN_{ijkl} / (\text{Máximo } (TPGFN_{ijkl}) - \text{Mínimo } (TPGFN_{ijkl}))) = r_{ij}$  prioridad de asignación de recursos normalizada entre valores extremos, como se puede ver en la Tabla 28.

Orden de asignación de los recursos	Prioridad de la asignación	Proceso al que se asignará el recurso
$r_{00}$	$T(EA;0.0000)$	$\beta_{00}$
$r_{01}$	$T(EA;-0.0326)$	$\beta_{01}$
$r_{11}$	$T(A;-0.0028)$	$\beta_{11}$
$r_{10}$	$T(A;-0.0355)$	$\beta_{10}$
$r_{20}$	$T(M;0.0423)$	$\beta_{20}$
$r_{21}$	$T(M;0.0337)$	$\beta_{21}$
$r_{30}$	$T(M;0.0274)$	$\beta_{30}$
$r_{31}$	$T(M;-0.0334)$	$\beta_{31}$
$r_{40}$	$T(B;-0.0581)$	$\beta_{40}$
$r_{41}$	$T(EB;0.0000)$	$\beta_{41}$

Tabla 28. Orden o prioridad final de asignación de los recursos y proceso al cual se asigna cada recurso en una iteración. Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas; cabe señalar que los recursos asignados estarán disponibles una vez que los procesos los liberen y, por lo tanto, podrán asignarse a otros procesos.

Dado que el sistema se autorregula liberando los recursos ya asignados a los procesos en el paso anterior, y dado que hay solicitudes de recursos de los procesos que aún no han sido satisfechas, los cálculos de las Tabla 27 y Tabla 28 se repiten con sus respectivos valores, omitiendo los procesos ya completados, como se puede observar en la Figura 9.



Figura 9. Proceso de autorregulación y cálculo de la FASDLN. Fuente: Elaboración propia.

El escenario que se presenta a continuación, parte de la concatenación de la asignación ordenada de cada una de las iteraciones correspondientes al escenario antes mencionado.

La tabla FASDLNC se obtendrá a partir de la concatenación de la tabla FASDLNO de cada iteración, como se muestra en la Tabla 29.





Una vez que se haya completado la tabla de la FASDLNC, se calcularán las Prioridades Finales Globales Lingüísticas del Proceso (PGFLP), para los cuales habrá que sumar el valor numérico representativo de cada 2-tupla de cada proceso y dividirlo por la cardinalidad de cada uno:

$$\begin{aligned}
 PGFLP_1 &= (T(B;-0.0581)+T(M;0.0813)+T(M;-0.0286)+T(EB;0)+T(MA;0.0431)+T(B;-0.0105)) / 6 \\
 PGFLP_2 &= (T(M;-0.0721) + T(B;0.0262) + T(EA;0) + T(M;-0.0467) + T(MA;-0.0133) + T(M;0.0638)) / 6 \\
 PGFLP_3 &= (T(M;0.0274) + T(A;-0.0892) + T(EA;0) + T(M;-0.0133) + T(M;-0.0769) + T(B;0.0227) + T(MA;0.0255) + T(M;-0.0178)) / 6 \\
 PGFLP_4 &= (T(M;0.081) + T(EB;0.0658) + T(B;-0.0337) + T(B;0.0275) + T(EA;0) + T(MA;-0.0348)) / 6 \\
 PGFLP_5 &= (T(M;-0.0187) + T(B;0.0674) + T(M;0.004) + T(EA;0)) / 4 \\
 PGFLP_6 &= (T(EA;0) + T(MA;0.0081) + T(EB;0) + T(A;-0.0738) + T(B;-0.0762)) / 5 \\
 PGFLP_7 &= (T(EB;0.0032) + T(EB;0) + T(MB;-0.0035) + T(EA;0)) / 4 \\
 PGFLP_8 &= (T(M;-0.0578)) / 1 \\
 PGFLP_9 &= (T(B;0.0169) + T(A;-0.0777) + T(EA;0)) / 3 \\
 PGFLP_{10} &= (T(EB;0.0468) + T(B;-0.0824) + T(MB;0.0273) + T(EA;0)) / 4 \\
 PGFLP_{11} &= (T(EB;0.023) + T(MB;0.0682) + T(MB;0.0031) + T(MB;-0.0553) + T(EA;-0.021) + T(MB;0.011) + T(MB;0.0076) + T(MA;-0.0117)) / 6 \\
 PGFLP_{12} &= (T(A;-0.0028) + T(M;0.0423) + T(M;-0.0034) + T(EB;0) + T(EA;0) + T(MA;0.0758) + T(M;-0.0047) + T(MB;0.0173)) / 6 \\
 PGFLP_{13} &= (T(EB;0) + T(MB;-0.0604) + T(MB;0.0742) + T(MB;0.035) + T(B;-0.0534) + T(MA;0.0734) + T(EA;0)) / 7 \\
 PGFLP_{14} &= (T(MB;-0.0681) + T(EB;0) + T(MB;0.0615) + T(MB;0.0443) + T(M;-0.0256) + T(MB;0.0181) + T(EA;0) + T(EB;0) + T(A;-0.0186)) / 8 \\
 PGFLP_{15} &= (T(EA;-0.0326) + T(A;-0.0358) + T(M;0.0337) + T(M;-0.0678) + T(EA;0)) / 5
 \end{aligned}$$

Al calcular el PGFLP para todos los procesos de la Tabla 30, se obtendrá un vector, como se muestra en la Tabla 31.

Prioridad Global Final	Procesos
T(M;-0.0782)	$\beta_{11}$
T(A;-0.0696)	$\beta_{12}$
T(A;-0.0717)	$\beta_{13}$
T(M;0.0178)	$\beta_{14}$
T(A;-0.0704)	$\beta_{15}$
T(M;0.0443)	$\beta_{16}$
T(B;-0.0417)	$\beta_{17}$
T(M;-0.0076)	$\beta_{18}$
T(A;-0.0208)	$\beta_{19}$
T(B;0.0401)	$\beta_{20}$
T(B;0.0108)	$\beta_{21}$
T(M;0.0289)	$\beta_{22}$
T(B;0.0607)	$\beta_{23}$
T(B;-0.0145)	$\beta_{24}$
T(A;0.0463)	$\beta_{25}$

Tabla 31. PGFLP (prioridad global final lingüística del proceso). Fuente: Elaboración propia.

Del vector PGFLP se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario global de asignación de recursos a procesos, como se puede observar en la Tabla 32.

Prioridad Global Final Ordenada	Procesos
T(A;0.0463)	$\beta_{25}$
T(A;-0.0208)	$\beta_{19}$
T(A;-0.0696)	$\beta_{12}$
T(A;-0.0704)	$\beta_{15}$
T(A;-0.0717)	$\beta_{13}$
T(M;0.0443)	$\beta_{16}$
T(M;0.0289)	$\beta_{22}$
T(M;0.0178)	$\beta_{14}$
T(M;-0.0076)	$\beta_{18}$
T(M;-0.0782)	$\beta_{11}$
T(B;0.0607)	$\beta_{23}$
T(B;0.0401)	$\beta_{20}$
T(B;0.0108)	$\beta_{21}$
T(B;-0.0145)	$\beta_{24}$
T(B;-0.0417)	$\beta_{17}$

Tabla 32. PGFLPD (prioridad global final lingüística del proceso ordenada). Fuente: Elaboración propia.



## Función de Asignación para Sistemas Distribuidos Lingüística Normalizada Concatenada Ordenada (FASDLNCO)

La FASDLNCO establecerá el orden de asignación por prioridades globales finales lingüísticas de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno, recibiendo todos los recursos solicitados. Para ello, se tendrá en cuenta la tabla FASDLNC y la PGFLPO.

Se calcularán las cardinalidades (número de asignaciones de recursos a cada proceso) obtenidas de cada uno de los procesos del vector PGFLPO en la tabla FASDLNC.

$cp_1$ = cardinalidad proceso(PGFLPO<sub>1</sub>) en FASDLNC  
 $cp_{21}$ = cardinalidad proceso(PGFLPO<sub>1</sub>) en FASDLNC = 5  
 $cp_{22}$ = cardinalidad proceso(PGFLPO<sub>2</sub>) en FASDLNC = 3  
 $cp_{12}$ = cardinalidad proceso(PGFLPO<sub>2</sub>) en FASDLNC = 6  
 $cp_{23}$ = cardinalidad proceso(PGFLPO<sub>3</sub>) en FASDLNC = 4  
 $cp_{13}$ = cardinalidad proceso(PGFLPO<sub>3</sub>) en FASDLNC = 8  
 $cp_{24}$ = cardinalidad proceso(PGFLPO<sub>4</sub>) en FASDLNC = 5  
 $cp_{14}$ = cardinalidad proceso(PGFLPO<sub>4</sub>) en FASDLNC = 6  
 $cp_{25}$ = cardinalidad proceso(PGFLPO<sub>5</sub>) en FASDLNC = 6  
 $cp_{15}$ = cardinalidad proceso(PGFLPO<sub>5</sub>) en FASDLNC = 1  
 $cp_{16}$ = cardinalidad proceso(PGFLPO<sub>6</sub>) en FASDLNC = 6  
 $cp_{26}$ = cardinalidad proceso(PGFLPO<sub>6</sub>) en FASDLNC = 7  
 $cp_{17}$ = cardinalidad proceso(PGFLPO<sub>7</sub>) en FASDLNC = 4  
 $cp_{27}$ = cardinalidad proceso(PGFLPO<sub>7</sub>) en FASDLNC = 8  
 $cp_{18}$ = cardinalidad proceso(PGFLPO<sub>8</sub>) en FASDLNC = 9  
 $cp_{28}$ = cardinalidad proceso(PGFLPO<sub>8</sub>) en FASDLNC = 4

Luego, habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla FASDLNC de cada uno de los procesos del vector PGFLPO. El total de elementos por cada proceso, estará determinado por la cardinalidad calculada en el paso anterior.

FASDLNCO<sub>1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>1</sub>) y FASDLNCO<sub>5</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>1</sub>).  
 FASDLNCO<sub>6-1</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>2</sub>) y FASDLNCO<sub>11-1</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>2</sub>)  
 FASDLNCO<sub>1-2-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>3</sub>)  
 FASDLNCO<sub>1-3-4</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>3</sub>)  
 FASDLNCO<sub>1-3-6-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>4</sub>) y FASDLNCO<sub>1-3-6-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>4</sub>)  
 FASDLNCO<sub>1-3-6-11-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>5</sub>) y FASDLNCO<sub>1-3-6-11-5</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>5</sub>)  
 FASDLNCO<sub>1-3-6-11-5-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>6</sub>) y FASDLNCO<sub>1-3-6-11-5-2</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>6</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>7</sub>) y FASDLNCO<sub>1-3-6-11-5-2-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>7</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-11-1</sub>: el único elemento de la FASDLNC para el proceso (PGFLPO<sub>8</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-11-2-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>9</sub>) y FASDLNCO<sub>1-3-6-11-5-2-11-2-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>9</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>10</sub>) y FASDLNCO<sub>1-3-6-11-5-2-11-2-11-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>10</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>11</sub>) y FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>11</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>12</sub>) y FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>12</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1-1-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>13</sub>) y FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1-1-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>13</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1-1-1-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>14</sub>) y FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1-1-1-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>14</sub>)  
 FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1-1-1-1-1</sub>: el primer elemento de la FASDLNC para el proceso (PGFLPO<sub>15</sub>) y FASDLNCO<sub>1-3-6-11-5-2-11-2-11-1-1-1-1-1-11</sub>: el último elemento de la FASDLNC para el proceso (PGFLPO<sub>15</sub>)



La Tabla 33 muestra el orden de todas las asignaciones de recursos para cada proceso, que es el primer proceso con mayor prioridad global y al que se asignan en primer lugar los recursos. La tabla completa continúa para cada una de las solicitudes de cada proceso (FASDLNCO).

Proceso	Asignación	Temática	Proceso	Asignación	Temática
TASA-00000	no asignado	1	TASA-00000	no asignado	7
TASA-00000	no asignado	1	TASA-00000	no asignado	8
TASA-00000	no asignado	1	TASA-00000	no asignado	9
TASA-00000	no asignado	1	TASA-00000	no asignado	10
TASA-00000	no asignado	1	TASA-00000	no asignado	11
TASA-00000	no asignado	1	TASA-00000	no asignado	12
TASA-00000	no asignado	1	TASA-00000	no asignado	13
TASA-00000	no asignado	1	TASA-00000	no asignado	14
TASA-00000	no asignado	1	TASA-00000	no asignado	15
TASA-00000	no asignado	1	TASA-00000	no asignado	16
TASA-00000	no asignado	1	TASA-00000	no asignado	17
TASA-00000	no asignado	1	TASA-00000	no asignado	18
TASA-00000	no asignado	1	TASA-00000	no asignado	19
TASA-00000	no asignado	1	TASA-00000	no asignado	20
TASA-00000	no asignado	1	TASA-00000	no asignado	21
TASA-00000	no asignado	1	TASA-00000	no asignado	22
TASA-00000	no asignado	1	TASA-00000	no asignado	23
TASA-00000	no asignado	1	TASA-00000	no asignado	24
TASA-00000	no asignado	1	TASA-00000	no asignado	25
TASA-00000	no asignado	1	TASA-00000	no asignado	26
TASA-00000	no asignado	1	TASA-00000	no asignado	27
TASA-00000	no asignado	1	TASA-00000	no asignado	28
TASA-00000	no asignado	1	TASA-00000	no asignado	29
TASA-00000	no asignado	1	TASA-00000	no asignado	30
TASA-00000	no asignado	1	TASA-00000	no asignado	31
TASA-00000	no asignado	1	TASA-00000	no asignado	32
TASA-00000	no asignado	1	TASA-00000	no asignado	33
TASA-00000	no asignado	1	TASA-00000	no asignado	34
TASA-00000	no asignado	1	TASA-00000	no asignado	35
TASA-00000	no asignado	1	TASA-00000	no asignado	36
TASA-00000	no asignado	1	TASA-00000	no asignado	37
TASA-00000	no asignado	1	TASA-00000	no asignado	38
TASA-00000	no asignado	1	TASA-00000	no asignado	39
TASA-00000	no asignado	1	TASA-00000	no asignado	40

Tabla 33. Orden Final de asignación en la FASDLNCO. Fuente: Elaboración propia.

De esta manera, se respondió a todas las solicitudes de recursos de todos los procesos, considerando la exclusión mutua y las prioridades de los procesos, las prioridades nodales y las prioridades finales, teniendo en cuenta los estrictos requisitos de consenso establecidos para este escenario.

### 6. Evaluación

El modelo propuesto logra establecer un consenso que permite a los procesos acceder a todos sus recursos de manera secuencial sin que éstos puedan ser removidos hasta que el proceso que los mantiene los libere. El orden de otorgamiento de recursos será determinado por la prioridad promedio general de todas las asignaciones en el supuesto del escenario general. El sistema distribuido regula y actualiza constantemente el estado local de cada nodo, las decisiones de acceso a los recursos modifican estos estados por lo que debe ser reajustado repetidamente, garantizando la exclusión mutua y reordenando nuevas prioridades. El método debe repetirse siempre que haya procesos que requieran recursos compartidos.

En La Red Martínez (2017), en el escenario general, las asignaciones de recursos a los diferentes procesos se realizan en varias rondas de asignación, donde un proceso puede tener varias asignaciones de recursos en las diferentes rondas.

Este capítulo considera el promedio global de prioridades que cada proceso tiene sobre todos los recursos de todas sus asignaciones en las diferentes rondas, pero para la asignación global final de cada proceso, respeta el mismo orden de asignación de cada recurso en las diferentes rondas en las que fueron asignados en el escenario general.



Es decir, la elección del proceso al cual se le otorga los recursos, se determina con el promedio global de prioridades en todas las asignaciones de todas las rondas PGFLPO, pero el orden en el que se deben realizar esas asignaciones, respeta el de la tabla FASDLNC, para cada proceso.

La representación de los  $r_{ij}$  indican los recursos (cuyo primer subíndice representa al nodo donde se encuentra y el segundo subíndice al propio número de recurso) que se le asignan al proceso  $p_{ek}$  (cuyo primer subíndice representa al nodo donde se encuentra y el segundo subíndice al propio número de proceso) en cada ronda, ver Figura 10.

Aunque para los recursos los subíndices son distintos, no necesariamente sean recursos distintos, sino que pueden representar a un mismo recurso que se asigna varias veces en las distintas rondas, pero siempre al mismo proceso  $p_{ek}$ . La ubicación en la tabla FASDLNCO dependerá de la ubicación en el vector PGFLPO.

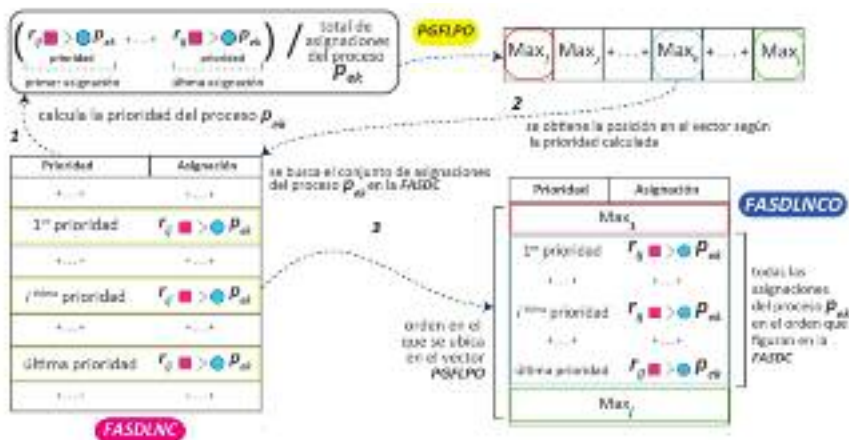


Figura 10. Cálculo de prioridades para el proceso  $p_{ek}$  en PGFLPO. Fuente: Elaboración propia.

El modelo propuesto incluye, como caso particular, un método que consiste en considerar la prioridad global de los procesos, en lugar de un grupo de variables de estado de cada nodo.

## 7. Discusiones y comentarios

Como los procesos se ejecutan en distintos procesadores y con todos sus recursos, no hay conflicto en cuanto a tener varios procesos en el mismo procesador.

En este escenario no se tiene en cuenta la cantidad de tiempo que cada proceso estará utilizando un nodo en particular. Tampoco se tiene en cuenta la cantidad de tiempo en que cada recurso estará asignado a un proceso en particular.

Se trata de un modelo general sin contemplar grupos de procesos. Se entiende por consenso que se priorice al proceso que resulte con mayor prioridad global promedio en sus asignaciones, según las tablas de asignaciones concatenada (FASDLNC).

Otra característica notable de la propuesta es su facilidad de implementación en el entorno de un administrador centralizado de recursos compartidos de un sistema distribuido.

Se buscó que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua pudiendo constituir grupos de procesos (los procesos independientes son considerados grupos unitarios); los procesos no requerían sincronización y tenían exigencias estrictas de consenso para lograr el acceso (se requería un consenso para asignar de manera consecutiva los recursos solicitados por un proceso o grupo de procesos, es decir, que iniciada la secuencia de asignación de recursos a un proceso, la misma no pudo ser interrumpida).



para asignar recursos a otro proceso). Se ha presentado una variante de un método innovador para la gestión de recursos compartidos en sistemas distribuidos, basado en La Red Martínez (2017), Agostini et al. (2018) y Agostini et al. (2019). Se ha establecido un modelo de consenso que favoreció el acceso secuencial de los procesos a todos los recursos solicitados. Se utilizó la tabla FASDLNC, que es la concatenación de las tablas FASDLNO de todas las rondas del escenario general y consideró el promedio global de prioridades que cada proceso tuvo sobre todos los recursos de todas sus asignaciones en las diferentes rondas.

## 8. Conclusiones

El modelo propuesto permite que el sistema distribuido se autorregule repetidamente según el estado local de los nodos, lo que resulta en una actualización de sus estados locales, como consecuencia de la evolución de sus respectivos procesos y de las decisiones de acceso a los recursos: el sistema distribuido en cuyos grupos de procesos se ejecuta el acceso a los recursos críticos, produce decisiones de acceso a los recursos que modifican el estado del sistema y lo reajustan repetidamente, garantizando además la exclusión mutua en el acceso a los recursos compartidos, indicando la prioridad de concesión de acceso a cada recurso y el proceso al que está asignado. Este proceso se repite si existen procesos que solicitan el acceso a los recursos compartidos.

El uso del modelo lingüístico de 2-tuplas permite mejorar la precisión y facilitar el procesamiento de textos tratando el dominio lingüístico como continuo, pero manteniendo la base lingüística (sintáctica y semántica), a través de la traducción simbólica.

Como línea futura, se prevé elaborar variantes del método propuesto teniendo en cuenta otros operadores de agregación (especialmente la familia OWA) y la posibilidad de ser utilizado por un gestor de recursos compartidos (en lugar de centralizado como en el método propuesto).

## Agradecimientos

Esta labor ha sido apoyada por el Proyecto: "Modelos de decisión y operadores de agregación para la gestión de procesos en sistemas distribuidos", código 16F001 de la Universidad Nacional del Nordeste (Argentina).

Cómo citar este artículo / How to cite this paper

Fomerón Martínez, J. T.; Agostini, F.; La Red Martínez, D. L. (2020). Gestión de procesos basado en lógica difusa con estrictos niveles de consenso. *International Journal of Information Systems and Software Engineering for Big Companies (IJISEBC)*, 7(1), 51-77. ([www.ijisebc.com](http://www.ijisebc.com))

## Referencias

- Agostini, F.; La Red Martínez, D. L.; Acosta, J. C. (2018). Modeling of the consensus in the allocation of resources in distributed systems. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 9(12). doi:10.14569/IJACSA.2018.091204.
- Agostini, F.; La Red Martínez, D. L.; Acosta, J. C., (2019). Assignment of Resources in Distributed Systems With Strict Consensus Requirements. In 10th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2019). Orlando, Florida, USA.
- Agrawal, D.; El Abbadi, A. (1991). An efficient and fault-tolerant solution for distributed mutual exclusion. *ACM Transactions on Computer Systems (TOCS)*, 9(1), 1-20.
- Andrews, G. (2000). *Foundation of Multithreaded, Parallel, and Distributed Programming*. Reading, MA: Addison Wesley.
- Attiya, H.; Welch, J. (2004). *Distributed computing: fundamentals, simulations, and advanced topics (Vol. 19)*. John Wiley & Sons.
- Cao, G.; Singhal, M. (2001). A Delay-Optimal Quorum-Based Mutual Exclusion Algorithm for Distributed Systems. *IEEE Transactions on Parallel and Distributed Systems*, 12(12), 1256-1268.
- Chen, B.; Wan, J.; Shu, L.; Li, P.; Mukherjee, M.; Yin, B. (2018). Smart Factory of Industry 4.0: Key Technologies, Application Case, and Challenges. *IEEE Access*, 6, 6505-6519. doi:10.1109/ACCESS.2017.2783682.
- Dell, M.; Fredman, C. (1999). *Direct from dell: Strategies that revolutionized an industry*: Harper Business.
- Dutta, B.; Labella, A.; Rodríguez, R. M.; Martínez, L. (2019). Aggregating Interrelated Attributes in Multi-Attribute Decision-Making With





- ELICIT Information Based on Bonferroni Mean and Its Variants. *International Journal of Computational Intelligence Systems*, 12(2), 1179-1196. doi:10.2991/ijcis.d.190930.002.
- Guerraoui, R.; Rodrigues, L. (2006). *Introduction to Reliable Distributed Programming*. Berlin: Springer-Verlag.
- Herrera, F.; Martínez, L. (2000). An Approach For Combining Linguistic And Numerical Information Based On The 2-Tupla Fuzzy Linguistic Representation Model In Decision-Making. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 8(5), 539-562.
- Jiménez, G. E.; Zulueta, Y. (2017). A 2-tupla linguistic multi-period decision making approach for dynamic green supplier selection. *DYNA*, 84(202), 199-206.
- Katzy, B. R.; Schuh, G. (1998). *The virtual enterprise Handbook of life cycle engineering: Concepts, Models and Technologies*. Springer.
- Khan, P. W.; Byun, Y. C.; Park, N. (2020). IoT-Blockchain Enabled Optimized Provenance System for Food Industry 4.0 Using Advanced Deep Learning. *Sensors*.
- La Red Martínez, D. L. (2017). Aggregation Operator for Assignment of Resources in Distributed Systems. (*IJACSA*) *International Journal of Advanced Computer Science and Applications*, 8(10).
- Lasi, H.; Fet-tke, P.; Kemper, H.-G.; Feld, T.; Hoffmann, M. (2014). *Industrie 4.0. Business & Information Systems Engineering*.
- Lau, H. C. W.; Chin, K. S.; Pun, K. F.; Ning, A. (2000). Decision supporting functionality in a virtual enterprise network. *Expert Systems with Applications*, 19(4), 261-270.
- Lin, S.-D.; Lian, Q.; Chen, M.; Zhang, Z. (2004). A practical distributed mutual exclusion protocol in dynamic peer-to-peer systems. In *International Workshop on Peer-to-Peer Systems* (pp. 11-21). Springer, Berlin, Heidelberg.
- Lodha, S.; Kshemkalyani, A. (2000). A fair distributed mutual exclusion algorithm. *IEEE Transactions on Parallel and Distributed Systems*, 11(6), 537-549.
- Liu, J.; Yi, L.; Pei, Z. (2018). A new linguistic term transformation method in linguistic decision making. *Journal of Intelligent & Fuzzy Systems*, 35, 2403-2412. doi:10.3233/JIFS-17987.
- Lynch, N. (1996). *Distributed Algorithms*. San Mateo, CA, USA: Morgan Kaufman.
- Mendel, J. M. (2007). Computing with words: Zadeh, turing, popper and occam. *IEEE computational intelligence magazine*, 2(4), 10-17.
- Ricart, G.; Agrawala, A. K. (1981). An optimal algorithm for mutual exclusion in computer networks. *Communications of the ACM*, 24(1), 9-17.
- Ríos, D. A.; La Red Martínez, D. L. (2019). Nuevo Modelo de Decisión Para Gestión de Tráfico en Redes. *International Journal Of Information Systems And software Engineering For Big Companies (IJISEBC)*, 6(2), 99-122.
- Saleem, H.; Bin Muhammad, K.; Saleem, S.; Saleem, R.; Hussain, A.; Aslam, A. M. (2019). Novel Intelligent Electronic Booking Framework for E-Business with Distributed Computing and Data Mining. *IJCSNS International Journal of Computer Science and Network Security*, 19(4), 270-278.
- Saxena, P.; Rai, J. (2003). A Survey of Permission-based Distributed Mutual Exclusion Algorithms. *Computer Standards and Interfaces*, (25)2, 159-181.
- Sha, L.; Rajkumar, R.; Lehoczky, J. P. (1990). Priority inheritance protocols: An approach to real-time synchronization. *IEEE Transactions on computers*, 39(9), 1175-1185.
- Stallings, W. (2005). *Sistemas Operativos*. España: Pearson Educación S. A.
- Tanenbaum, A. S. (1996). *Sistemas Operativos Distribuidos*. México: Prentice - Hall Hispanoamericana.
- Tanenbaum, A. S. (1999). *Sistemas Operativos Modernos*. México: Pearson Educación.
- Tel. G. (2000). *Introduction to Distributed Algorithms*. Cambridge, UK: Cambridge University Press.
- Velazquez, M. (1993). *A survey of distributed mutual exclusion algorithms*. Colorado State Univ.
- Wan, J.; Yi, M.; Li, D. I.; Zhang, C.; Wang, S.; Zhou, K. (2016). Mobile Services for Customization Manufacturing Systems: An Example of Industry 4.0. *IEEE Access*.
- Wan, J.; Cai, H.; Zhou, K. (2015). Industry 4.0: Enabling technologies. In *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*.
- Wen, T.; Chang, K.; Lai, H. (2020). Integrating the 2-tupla linguistic representation and soft set to solve supplier selection problems with incomplete information. *Engineering Applications of Artificial Intelligence*.
- Yager, R. (1988). On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1), 183-190.
- Yager, R.; Pasi, G. (2002). Modelling Majority Opinion in Multi-Agent Decision Making. *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*.
- Ying, M. (2002). A Formal Model of Computing With Words. *IEEE Transactions On Fuzzy Systems*, 10(5).
- Zapata, S.; Fuentealba, D.; Valenzuela, G. (2015). Aplicación del modelo de representación de información lingüística 2-tuplas con información multigranular. *Revista Trilogía: Ciencia, Tecnología y Sociedad*, 27(37), 110-127.

