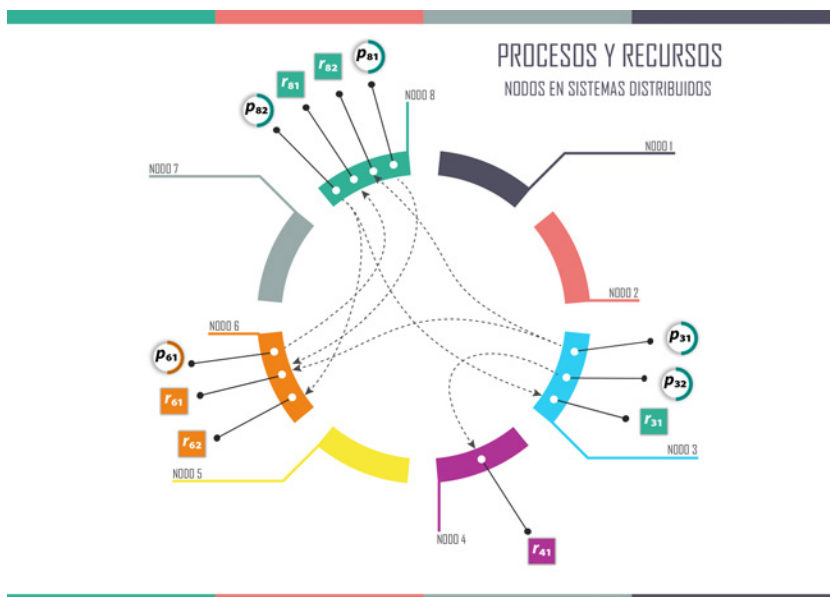




Universidad Nacional del Nordeste

Facultad de Ciencias Exactas y Naturales y Agrimensura

Maestría en Sistemas y Redes de Telecomunicaciones



Tesis de Maestría en Sistemas y Redes de Telecomunicaciones

TÍTULO

“Nueva propuesta para la administración de recursos y procesos en sistemas distribuidos”

Autor: **Lic. Federico Agostini**

Director: **Dr. David Luis la Red Martínez**

Año: **2019**

Dedicatoria

Al Dr. David Luis La Red Martínez, por sus consejos, el apoyo, el ánimo, la paciencia y dirección que me brindó durante la realización del trabajo.

A mi compañero y amigo Alberto Rios, con el que inicié la carrera, por darme fuerzas cuando las cosas se ponían complicadas, por ser ejemplo de superación y esfuerzo para afrontar los momentos difíciles de la vida.

Agradecimiento

Agradezco a las personas que me han ayudado a cumplir mis sueños, por motivarme y darme la mano cuando sentía que el camino se terminaba, una mención especial a Antonela, a mi familia y amigos, y a mis padres que me han formado como persona de bien y me han preparado para los retos de la vida, gracias de corazón.

Una mención muy especial al Dr. David Luis La Red Martínez, la Dra. Laura Leiva y la Cra. Laura Sanchez, gracias por sus ganas de transmitirme sus conocimientos y dedicación, personas de gran sabiduría quienes se han esforzado por ayudarme a llegar al punto en el que me encuentro.

Índice

Dedicatoria	i
Agradecimiento.....	ii
Índice.....	iii
Listado de tablas	v
Lista de figuras	viii
Resumen	ix
Capítulo I - Introducción	1
1.1. Introducción	1
1.2. Hipótesis y objetivos	2
1.3. Antecedentes.....	4
1.4. Marco teórico.....	5
1.5. Metodología.....	11
1.6. Estructura de la tesis.....	15
1.7. Discusiones y comentarios.....	17
Capítulo II – Operador de agregación definido para el escenario 1 (E1)	18
2.1. Introducción	18
2.2. Hipótesis y objetivos	18
2.3. Propuesta de solución.....	18
2.4. Descripción del operador de agregación.....	22
2.5. Ejemplo	29
2.6. Evaluación.....	41
2.7. Discusiones y comentarios.....	41
Capítulo III - Operador de agregación definido para el escenario 2 (E2)	43
3.1. Introducción	43
3.2. Hipótesis y objetivos	43
3.3. Propuesta de solución.....	43
3.4. Descripción del operador de agregación.....	45
3.5. Ejemplo	51
3.6. Evaluación.....	62
3.7. Discusiones y comentarios.....	64
Capítulo IV - Operador de agregación definido para el escenario 3 (E3).....	66
4.1. Introducción	66
4.2. Hipótesis y objetivos	66
4.3. Propuestas de solución.....	66
4.4. Descripción del operador de agregación.....	68
4.5. Ejemplo	78
4.6. Evaluación.....	96
4.7. Discusiones y comentarios.....	98
Capítulo V - Operador de agregación definido para el escenario 4 (E4).....	100

5.1.	Introducción	100
5.2.	Hipótesis y objetivos	100
5.3.	Propuestas de solución	100
5.4.	Descripción del operador de agregación.....	101
5.5.	Ejemplo	106
5.6.	Evaluación	130
5.7.	Discusiones y comentarios.....	132
Capítulo VI - Modelos de decisión y operadores de agregación		133
6.1.	Introducción a los modelos de decisión	133
6.2.	Modelo de decisión propuesto.....	133
6.3.	Consideraciones del modelo de decisión propuesto	134
6.1.	Ejemplo de modelo de decisión aplicado a alguno de los algoritmos tradicionales.....	135
6.2.	Consideraciones acerca de las operaciones de agregación	147
6.3.	Discusiones y comentarios.....	148
Capítulo VII - Conclusiones y futuras líneas de investigación		150
7.1.	Conclusiones	150
7.2.	Futuras líneas de investigación	153
Bibliografía.....		155
Apéndice de publicaciones		165

Listado de tablas

TABLA 1. Procesos en cada nodo.....	19
TABLA 2. Recursos compartidos disponibles en cada nodo.....	20
TABLA 3. Recursos solicitados por los procesos	20
TABLA 4. Criterios para medir la carga computacional en cada nodo	23
TABLA 5. Categorías para medir la carga computacional en cada nodo	23
TABLA 6. Criterios para calcular la prioridad o preferencia de cada nodo	24
TABLA 7. Pesos asignados a los criterios de cálculo de preferencias.....	25
TABLA 8. Valoraciones asignadas a los criterios para el cálculo de prioridades	26
TABLA 9. Prioridades nodales de los procesos para acceder a cada recurso	27
TABLA 10. Pesos asignados a los procesos para el cálculo de prioridades	28
TABLA 11. Pesos finales normalizados asignados a los procesos.....	28
TABLA 12. Orden o prioridad final de asignación de los recursos a procesos	29
TABLA 13. Procesos en cada nodo.....	30
TABLA 14. Procesos en cada grupo	30
TABLA 15. Recursos compartidos disponibles en cada nodo.....	31
TABLA 16. Recursos solicitados por los procesos	31
TABLA 17. Valores de los criterios de carga computacional.....	31
TABLA 18. Valores de las categorías para medir la carga computacional.....	32
TABLA 19. Pesos asignados a los criterios para calcular la prioridad	33
TABLA 20. Valoraciones asignadas a los criterios para el cálculo de prioridad	33
TABLA 21. Prioridades globales finales para asignar los recursos.....	37
TABLA 22. Orden o prioridad final de asignación	37
TABLA 23. Prioridades nodales de los procesos para acceder a cada recurso	38
TABLA 24. Pesos finales y pesos finales normalizados	38
TABLA 25. Prioridades globales finales de los procesos para acceder a cada recurso	38
TABLA 26. Orden o prioridad final (segunda iteración).....	39
TABLA 27. Orden o prioridad final de asignación (tercera iteración)	39
TABLA 28. Orden o prioridad final de asignación (cuarta iteración).....	39
TABLA 29. Orden o prioridad final de asignación (quinta iteración).....	39
TABLA 30. Orden o prioridad final de asignación (sexta iteración).....	40
TABLA 31 . Orden o prioridad final de asignación (séptima iteración)	40
TABLA 32. Orden o prioridad final de asignación (octava iteración).....	40
TABLA 33. Orden o prioridad final de asignación (novena iteración).....	40
TABLA 34. Orden o prioridad final de asignación (décima iteración).....	40
TABLA 35. Orden o prioridad final de asignación (décimo primera iteración)	41
TABLA 36. Orden o prioridad final de asignación (décimo segunda iteración)	41
TABLA 37. Concatenación de las tablas de asignación ordenada (FASDC)	47
TABLA 38. Prioridad final global (FASD) primera iteración	52
TABLA 39. Orden final de asignación (FASDO) primer iteracción.....	52
TABLA 40. Orden final de asignación todas las iteraciones (FASDC).....	53
TABLA 41. Prioridad Global Final Ordenada por proceso	55
TABLA 42. PGFP (prioridad global final del proceso).....	57
TABLA 43. PGFPO (prioridad global final del proceso ordenada)	57
TABLA 44. Orden Final de asignación en la FASDCO	61
TABLA 45. Esquema general de rondas y de subrondas de cada grupo.....	77
TABLA 46. Orden Final de asignación en la FASDCO	78
TABLA 47. PGG (prioridad global final del grupo)	80
TABLA 48. PGGO (prioridad global final del grupo ordenada).....	80
TABLA 49. Primera subbronda de asignación compatible para el primer grupo (g_4).....	83
TABLA 50. Segunda subbronda de asignación compatible para el primer grupo (g_4).....	84
TABLA 51. Tercer subbronda de asignación compatible para el primer grupo (g_4).....	85
TABLA 52. Primera subbronda de asignación compatible para el segundo grupo (g_1)... ..	86
TABLA 53. Segunda subbronda de asignación compatible para el primer grupo (g_1).....	87

TABLA 54.	Tercer subbronda de asignación compatible para el primer grupo (g_1).....	87
TABLA 55.	Primera subbronda de asignación compatible para el tercer grupo (g_2)	88
TABLA 56.	Segunda subbronda de asignación compatible para el tercer grupo (g_2).....	89
TABLA 57.	Primera subbronda de asignación compatible para el cuarto grupo (g_3).....	90
Tabla 58.	Segunda subbronda de asignación compatible para el cuarto grupo (g_3).....	91
TABLA 59.	Primera subbronda de asignación compatible para el quinto grupo (g_0).....	92
TABLA 60.	Segunda subbronda de asignación compatible para el quinto grupo (g_0)	93
TABLA 61.	Tercer subbronda de asignación compatible para el quinto grupo (g_0)	94
TABLA 62.	Cuarta subbronda de asignación compatible para el quinto grupo (g_0).....	94
TABLA 63.	Orden Final de asignación en la FASDOGC	95
TABLA 64.	Esquema general de rondas y de subbrondas de cada grupo.....	106
TABLA 65.	Orden de asignación en la FASDO	107
TABLA 66.	PGA (prioridad de grupo en la asignación).....	107
TABLA 67.	PGA (prioridad de grupo en la asignación ordenada).....	107
TABLA 68.	Orden de asignación de cada grupo en la FASDOG (primer iteracion)	108
TABLA 69.	Orden de asignación por proceso en la FASDO	109
TABLA 70.	PGA (prioridad de grupo en la asignación).....	110
TABLA 71.	PGA (prioridad de grupo en la asignación ordenada)	110
Tabla 72.	Orden de asignación de grupo en la FASDOG (segunda asignación)	111
TABLA 73 .	Orden de asignación por proceso en la FASDO	111
TABLA 74 .	PGA (prioridad de grupo en la asignación).....	112
TABLA 75.	PGA (prioridad de grupo en la asignación ordenada).....	112
TABLA 76.	Orden de asignación de grupo en la FASDOG (tercer asignación)	113
TABLA 77.	Orden de asignación en la FASDO	114
TABLA 78.	PGA (prioridad de grupo en la asignación).....	114
TABLA 79.	PGA (prioridad de grupo en la asignación ordenada)	115
TABLA 80.	Orden de asignación de grupo en la FASDOG (cuarta asignación).....	116
TABLA 81.	Orden de asignación en la FASDO	116
TABLA 82.	PGA (prioridad de grupo en la asignación).....	117
TABLA 83.	PGA (prioridad de grupo en la asignación ordenada)	117
TABLA 84.	Orden de asignación de grupo en la FASDOG (quinta asignación).....	118
TABLA 85.	Orden de asignación en la FASDO	119
TABLA 86.	PGA (prioridad de grupo en la asignación).....	120
TABLA 87 .	PGA (prioridad de grupo en la asignación ordenada)	120
TABLA 88.	Orden de asignación de grupo en la FASDOG (sexta asignación)	121
TABLA 89.	Orden de asignación en la FASDO	121
TABLA 90.	PGA (prioridad de grupo en la asignación).....	122
TABLA 91.	PGA (prioridad de grupo en la asignación ordenada)	122
TABLA 92.	Orden de asignación de grupo en la FASDOG (séptima asignación)	123
TABLA 93.	Orden de asignación en la FASDO	124
TABLA 94.	PGA (prioridad de grupo en la asignación).....	124
TABLA 95.	PGA (prioridad de grupo en la asignación ordenada)	124
TABLA 96.	Orden de asignación de grupo en la FASDOG (octava asignación)	125
TABLA 97.	Orden de asignación en la FASDO	126
TABLA 98.	PGA (prioridad de grupo en la asignación).....	126
TABLA 99.	PGA (prioridad de grupo en la asignación ordenada)	126
TABLA 100.	Orden de asignación de grupo en la FASDOG (novena asignación)	127
TABLA 101.	Orden de asignación en la FASDO	127
TABLA 102.	PGA (prioridad de grupo en la asignación).....	128
TABLA 103.	PGA (prioridad de grupo en la asignación ordenada)	128
TABLA 104.	Orden de asignación de grupo en la FASDOG (décima asignación)	129
TABLA 105.	Orden de asignación en la FASDO	129
TABLA 106.	PGA (prioridad de grupo en la asignación).....	129
TABLA 107 .	PGA (prioridad de grupo en la asignación ordenada)	129
TABLA 108.	Orden de asignación de grupo en la FASDOG (undécima asignación) ..	130

TABLA 109. Pesos asignados a los procesos para el cálculo de prioridades	135
TABLA 110. Pesos asignados a los criterios para calcular la prioridad	136
TABLA 111. Valoraciones asignadas a los criterios para el cálculo de prioridad	136
TABLA 112. Prioridades nodales de los procesos para acceder a cada recurso	140
TABLA 113. Pesos finales y pesos finales normalizados	140
TABLA 114. Prioridades globales finales de los procesos para acceder a cada recurso	140
TABLA 115. Cálculo de prioridades de asignación de recursos	141
TABLA 116. Cálculo de prioridades de asignación de recursos ordenada	141
TABLA 117. Orden o prioridad final de asignación (FASD)	142
TABLA 118. Orden o prioridad final de asignación	142
TABLA 119. Orden o prioridad final de asignación (décima iteración)	143
TABLA 120. Orden o prioridad final de asignación (sexta iteración)	143
TABLA 121. Orden o prioridad final de asignación (tercera iteración)	143
TABLA 122. Orden o prioridad final de asignación (cuarta iteración)	144
TABLA 123. Orden o prioridad final de asignación (cuarta iteración)	144
TABLA 124. Orden o prioridad final de asignación (quinta iteración)	144
TABLA 125. Orden o prioridad final de asignación (sexta iteración)	145
TABLA 126. Orden o prioridad final de asignación (quinta iteración)	145
TABLA 127. Orden o prioridad final de asignación (quinta iteración)	145
TABLA 128. Cuadro comparativo del modelo propuesto y los tradicionales	146

Lista de figuras

Fig. 1. Recursos y procesos en nodos en sistemas distribuidos	45
Fig. 2. Pasos para obtener las funciones FASD , FASDO y FASDC	46
Fig. 3. Pasos para la FASD , FASDO y FASDC en cada iteración.	47
Fig. 4. Cálculo de la PGFP de cada proceso	48
Fig. 5. Ejemplo para caclular la PGFPO de cada proceso	49
Fig. 6. Pasos para obtener desde la FASDC a la FASDCO	49
Fig. 7. Esquema para obtener la FASDCO de cada proceso	50
Fig. 8. Cálculo de prioridades para el proceso p_{ek} con mayor prioridad en PGFPO	63
Fig. 9. Pasos para obtener desde la FASDCO a la FASDOGC	69
Fig. 10. Cálculo de la PGG de cada grupo	69
Fig. 11. Ejemplo para calcular la PGGO de cada grupo de procesos	70
Fig. 12. Listado final del grupo de mayor prioridad en la PGGO	98
Fig. 13. Pasos para obtener desde la FASDO a la FASDOG	102
Fig. 14. Cálculo de la PGA de cada grupo en la asignación	103
Fig. 15. Ejemplo para calcular la PGAO de cada grupo de procesos	104
Fig. 16. Cálculo desde la FASDO a la FASDOG en la primera ronda	131
Fig. 17. Modelo global de decisión para acceso a recursos compartidos	134
Fig. 18. Modelo de decisión propuesto	134
Fig. 19. Comparación del Modelo propuesto y los modelos tradicionales.....	135
Fig. 20. Modelo global de decisión para accesos a recursos compartidos.	146
Fig. 21. Modelo de decisión incluyendo en el proceso de resolución los métodos de agregación específicos para al acceso a recursos compartidos.	147

Resumen

Esta Tesis describe la forma en que se pueden utilizar nuevos modelos para la toma de decisiones en grupos de procesos distribuidos, que requieren acceso a recursos compartidos en la modalidad de exclusión mutua.

Los modelos de decisión actualmente disponibles y generalmente aplicables en los sistemas distribuidos se basan en algoritmos de intercambio de permisos que intentan lograr un acuerdo de todos los procesos intervinientes para realizar determinadas acciones.

Una propuesta interesante, es la creación de un modelo de decisión que permite adaptarse a diferentes requerimientos, que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua, que requieran o no sincronización, que puedan pertenecer a grupos de procesos y que puedan tener exigencias estrictas de consenso para lograr el acceso a los recursos, y en base a esa información el modelo pueda elegir el operador de agregación correspondiente a cada caso.

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 del 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 existe un *Runtime* coordinador global en uno de los nodos que evalúa y ejecuta el modelo de decisión y el operador de agregación correspondiente.

Los escenarios propuestos en este documento consideran los recursos y procesos de los sistemas operativos distribuidos, aplicados al entorno de las telecomunicaciones, pero sin limitarse a ningún protocolo de comunicaciones específico, se trata de un esquema genérico. Se considera que la aplicación del método propuesto supondría un aumento del tráfico de información de control, pero el rendimiento global del sistema mejoraría al asignar recursos a los procesos de acuerdo con un esquema de toma de decisiones holístico y cognitivo que también garantiza la exclusión mutua en el acceso a los recursos compartidos.

Palabras clave: Sistemas operativos - Comunicación entre grupos de procesos - Operadores de agregación-Modelo de decisión-Consenso

Abstract

This thesis describes how new models can be used for decision making in distributed process groups that require access to shared resources in a mutually exclusive mode.

The decision models currently available and generally applicable in distributed systems are based on permission exchange algorithms that attempt to achieve an agreement of all intervening processes to perform certain actions.

An interesting proposal is the creation of a decision model that allows adapting to different requirements, that processes access shared resources in the mutual exclusion modality, that require or not synchronization, that may belong to groups of processes and that may have strict consensus requirements to achieve access to resources, and based on that information the model may choose the operator of aggregation corresponding to each case.

At each node, an interface is defined between the applications and the operating system, which through a *Runtime* (software in execution time complementary to the operating system) included in that interface, manages the shared processes and resources and defines the corresponding scenario. In addition, the *Runtime* interact with each other to exchange information and there is a global *Runtime* coordinator in one of the nodes that evaluates and executes the decision model and the corresponding aggregation operator.

The scenarios proposed in this document consider the resources and processes of the distributed operating systems, applied to the telecommunications environment, but not limited to any specific communications protocol, it is a generic scheme. It is considered that the application of the proposed method would lead to an increase in control information traffic, but the overall performance of the system would be improved by allocating resources to processes according to a holistic and cognitive decision-making scheme that also guarantees mutual exclusion in access to shared resources.

Keywords: Operating systems - Communication between groups of processes - Aggregation operators - Decision model - Consensus

Capítulo I - Introducción

1.1. Introducción

La proliferación de sistemas informáticos, muchos de ellos distribuidos, en los cuales existen múltiples procesos que cooperan para el logro de una determinada función, hace necesario disponer de modelos de decisión que permitan a los procesos intervinientes en los distintos grupos de procesos, tomar decisiones en las que son necesarios diferentes niveles de acuerdo, especialmente cuando se trata del acceso a recursos computacionales compartidos y el sistema debe auto-regular la forma de dicha compartición. Es especialmente significativo el caso del acceso a las llamadas regiones críticas de memoria por parte de distintos procesos, que pueden estar operando en equipos distribuidos, donde el acceso a las regiones críticas debe hacerse en la modalidad de acceso exclusivo y con el consentimiento de los demás procesos del grupo. Ejemplos de lo mencionado se encuentran en (Tanenbaum, 1996 y 2009), donde se describen los principales algoritmos de sincronización en sistemas distribuidos, en (Agrawal et al., 1991), donde se presenta una solución eficiente y tolerante a fallas para el problema de la exclusión mutua distribuida, en (Ricart et al., 1981), (Cao y Singhal, 2001) y en (Lodha y Kshemkalyani, 2000), donde se presentan unos algoritmos para gestionar la exclusión mutua en redes de computadoras, en (La Red Martínez, 2004), donde se describen los principales algoritmos de sincronización en sistemas distribuidos, en (Stallings, 2005), donde se detallan los principales algoritmos para la gestión distribuida de procesos, los estados globales distribuidos y la exclusión mutua distribuida. Estos temas y otros relacionados también han sido tratados en (Joshi y Holzmann, 2007), (Alagarsamy, 2003), etc.

Los modelos de decisión actualmente disponibles y generalmente aplicables en los sistemas distribuidos se basan en algoritmos de intercambio de permisos que intentan lograr un acuerdo de todos los procesos intervinientes para realizar determinadas acciones, como el acceso a un área de memoria compartida a la que se debe acceder en

la modalidad de exclusión mutua. Se considera especialmente importante estudiar la aplicación de modelos de decisión para la toma de decisiones en grupo, que dichos grupos de procesos mejorarían su desempeño mediante los modelos de decisión que se tiene previsto desarrollar incorporando mecanismos de auto-regulación. Para ello, se han generado nuevos modelos de toma de decisiones en grupos de procesos distribuidos.

1.2. Hipótesis y objetivos

Hipótesis

Desarrollar modelos de decisión para la toma de decisiones en grupos de procesos, utilizando la familia de operadores *OWA*, generando operadores específicos para cada uno de los siguientes tipos de situaciones:

- Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua sin constituir grupos de procesos que requieran sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y sin exigencias estrictas de consenso para lograr el acceso.

- Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua sin constituir grupos de procesos que requieran sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y con exigencias estrictas de consenso para lograr el acceso.

- Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua constituyendo grupos de procesos que requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y con exigencias estrictas de consenso para lograr el acceso.

- Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua constituyendo grupos de procesos que requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y sin exigencias estrictas de consenso para lograr el acceso.

Objetivo general

* Generar modelos de decisión innovadores y sus correspondientes operadores de agregación para la gestión de grupos de procesos.

Objetivos específicos

* Generar modelos de decisión y sus correspondientes operadores de agregación para la gestión de grupos de procesos que pueden compartir recursos, estudiando la utilización de posibles modificaciones de los operadores de la familia OWA (Yager, 1988, 1993) y la creación de operadores nuevos, para los siguientes tipos de situaciones:

** Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua sin constituir grupos de procesos que requieran sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y sin exigencias estrictas de consenso para lograr el acceso.

** Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua sin constituir grupos de procesos que requieran sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y con exigencias estrictas de consenso para lograr el acceso.

** Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua constituyendo grupos de procesos que requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y con exigencias estrictas de consenso para lograr el acceso.

** Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua constituyendo grupos de procesos que requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y sin exigencias estrictas de consenso para lograr el acceso.

* Comparar los modelos que se desarrollaron con los modelos conocidos y generalmente aceptados de las ciencias de la computación.

1.3. Antecedentes

Se han realizado y publicado varios trabajos relacionados con las áreas del conocimiento comprendidas en el presente proyecto; dichos trabajos se agrupan según se indica a continuación:

Cibernética

En (Castillo et al. 2005) se presenta una solución óptima para las operaciones de búsqueda en disco por parte de los sistemas operativos.

En (La Red Martínez, 2010a, 2010b) se analizan las principales concepciones acerca de la mente y de una concepción integrada acerca de la misma, estudiándose además si la mente podría ser considerada como un artefacto, discutiéndose las principales tendencias al respecto.

En (La Red et al., 2002) se estudia la optimización de operaciones de búsqueda en disco mediante redes neuronales artificiales.

Comunicación en sistemas distribuidos

En (La Red Martínez, 2004) se describen los principales algoritmos de comunicación en sistemas distribuidos (algoritmos clásicos de las ciencias de la computación).

En (La Red Martínez y Agostini, 2013, 2014) se presenta un entorno de enseñanza – aprendizaje de las comunicaciones de datos mediante animaciones.

Sincronización en sistemas distribuidos

En (La Red Martínez, 2004) se describen los principales algoritmos de sincronización en sistemas distribuidos (algoritmos clásicos de las ciencias de la computación).

Sistemas de soporte de decisión

En (Doña et al., 2011) se presenta un modelo de decisión en grupo con la

utilización de operadores de agregación de la familia OWA (Ordered Weighted Averaging: Promedio de Pesos Ordenados).

En (La Red Martínez y Acosta, 2015) se presentan las principales propiedades matemáticas y las medidas de comportamiento relacionadas con los operadores de agregación.

En (La Red Martínez y Pinto, 2015) se presenta una revisión acerca de los operadores de agregación, especialmente los de la familia OWA.

En (Peláez et al., 2003, 2004, 2009) se analizan operadores de agregación (de mayoría) en grupo que buscan la representación de la mayoría.

Toma de decisiones en grupo

En (La Red et al., 2011a) se presenta el operador *WKC-OWA* para agregar información en problemas de decisión democrática.

En (La Red Martínez y Acosta, 2014) y en (La Red Martínez et al., 2014) se presenta una nueva perspectiva para los modelos de decisión para la sincronización de procesos en sistemas distribuidos.

En (La Red Martínez y Acosta, 2015) se realiza una revisión del modelado de preferencias para los modelos de decisión.

1.4. Marco teórico

Se detallan a continuación los principales aportes al estado del conocimiento en las distintas áreas relacionadas con la presente tesis.

Cibernética

En (Bateson, 1991) se expresa la importancia del contexto en la comunicación. Toda comunicación exige un contexto, porque sin contexto no hay significado, no hay valor diferencial que genere información. Se presenta una visión sistémica e interdisciplinaria de los procesos comunicativos, que concibe los procesos comunicacionales con un

carácter circular y evolutivo, donde el feedback tiene una importancia decisiva.

En (Castillo et al. 2005) se presenta una solución óptima para las operaciones de búsqueda en disco por parte de los sistemas operativos.

En (Gardner, 1996) se expresa que el pensamiento humano resultará a la postre semejante, en aspectos significativos, a las operaciones de la computadora, y en particular a las de la computadora electrónica digital secuencial, aunque no se sepa en qué medida los procesos del pensamiento humano son computacionales. Se considera que es válido poner en tela de juicio que los procesos superiores que consideramos más exclusivamente humanos, puedan abordarse adecuadamente mediante el modelo computacional mencionado. A esto se denomina la paradoja computacional y significa que la aplicación de los métodos y modelos computacionales ha llevado a comprender en qué aspectos los seres humanos no se asemejan a las computadoras.

En (La Red et al., 2002) se estudia la optimización de operaciones de búsqueda en disco mediante redes neuronales artificiales.

En (Parsegian, 1973), (Wiener, 1961), (Brix, 1970) y (Longo, 1972) se explican los principales fundamentos de la cibernética.

Asimismo, el modelo de comunicación de (Shannon y Weaver, 1949) consta de varios componentes: una fuente de información que produce un mensaje o información que será transmitida; un transmisor que convierte el mensaje en señales electrónicas o electromagnéticas; estas señales son transmitidas a través de un canal o medio, que es el tercer componente; el cuarto componente es el receptor, que transforma de nuevo la señal recibida en el mensaje original; y el último componente es el destinatario, quien recibe el mensaje.

En (Tenenbaum et al., 2006), se introduce un framework basado en teoría bayesiana para modelar el aprendizaje inductivo y el razonamiento como inferencias estadísticas sobre representaciones estructuradas del conocimiento.

En (Wiener, 1985) se definen los principales conceptos de la cibernética; se expresa que la cibernética es todo campo de la teoría del mando y de la comunicación, tanto en la máquina, como en el animal. También afirma que la cibernética procura hallar los elementos comunes al funcionamiento de las máquinas automáticas y al sistema nervioso del hombre, y desarrollar una teoría que abarque todo el campo del control y de la comunicación en las máquinas y en los organismos vivientes.

En (Varela, 1990) se presentan los principales conceptos de enacción y emergencia, y una reseña de la evolución de los conceptos centrales de la cibernética.

Comunicación en sistemas distribuidos

En (Birman et al., 1987) se presenta un protocolo de comunicación confiable en presencia de fallas, desde la óptica de las ciencias de la computación.

En (Birman et al., 1991) se presenta un protocolo de comunicación multicast para grupos de procesos en la modalidad atómica, desde la óptica de las ciencias de la computación.

En (Birman, 2005) se estudian tecnologías, servicios web y aplicaciones de sistemas distribuidos confiables.

En (Joseph et al., 1989) se presentan protocolos confiables de comunicaciones en la modalidad de broadcast, desde la óptica de las ciencias de la computación.

En (Kaashoek, 1992) se estudia en profundidad la comunicación entre grupos de procesos analizándose protocolos como el *FLIP*: Fast Local Internet Protocol y el *BP*: Broadcast Protocol, analizando la comunicación entre grupos confiable y eficiente, la programación paralela, la programación tolerante a fallos utilizando broadcasting y planteando una arquitectura de tres niveles, el inferior para el protocolo *FLIP*, el medio para la comunicación en grupos de procesos y el superior para las aplicaciones. Información adicional puede encontrarse en (van Renesse et. al., 2003).

En (La Red Martínez, 2004) se describen los principales algoritmos de

comunicación en sistemas distribuidos (algoritmos clásicos de las ciencias de la computación).

En (Macedonia et al., 1995) se describe una arquitectura de red para entornos virtuales de gran escala para soportar la comunicación entre grupos de procesos distribuidos.

En (Silberschatz et al., 2006) se presentan los principales algoritmos de coordinación distribuida y gestión de la exclusión mutua (algoritmos clásicos de las ciencias de la computación).

En (Tanenbaum, 1996 y 2009) y en (Tanenbaum y Van Steen, 2008) se describen los principales algoritmos de comunicación en sistemas distribuidos (algoritmos clásicos de las ciencias de la computación).

Metodología

En (Samaja, 1993, 2004) y en (Ynoub, 2007) se analiza una importante metodología de investigación científica, en la que se destaca la importancia otorgada al proceso de investigación como un todo que incluye al diseño del objeto modelo de investigación y al desarrollo del proyecto de investigación, resaltándose la importancia conferida a la correcta especificación de la unidad de análisis, las dimensiones de análisis o variables, los valores o categorías, los indicadores o definiciones operacionales (entificar, clasificar, operacionalizar), integrando todo ello en los sistemas de matrices de datos.

Sincronización en sistemas distribuidos

En (Agrawal et al., 1991) se presenta una solución eficiente y tolerante a fallas para el problema de la exclusión mutua distribuida, desde la óptica de las ciencias de la computación.

En (Alagarsamy, 2003) se analizan los principales algoritmos de exclusión mutua.

En (Bagrodia, 1989) se analiza el diseño y la evaluación de rendimiento de

algoritmos distribuidos para la sincronización de procesos. Se presenta una solución sencilla para el problema de coordinación de comité, que abarca los problemas de sincronización y exclusión asociados con la implementación de encuentro de múltiples vías.

En (Joshi y Holzmann, 2007) se analiza la verificabilidad en un sistema de archivos.

En (La Red Martínez, 2004) se describen los principales algoritmos de sincronización en sistemas distribuidos (algoritmos clásicos de las ciencias de la computación).

En (Lamport, 1978) se estudia el tiempo, los relojes y el ordenamiento de eventos en sistemas distribuidos; se analiza el ordenamiento parcial, los relojes lógicos, el ordenamiento total de eventos, el comportamiento anormal, los relojes físicos, etc.

En (Ricart et al., 1981), (Cao y Singhal, 2001) y en (Lodha y Kshemkalyani, 2000) se presentan unos algoritmos para gestionar la exclusión mutua en redes de computadoras, conforme a las ciencias de la computación.

En (Stallings, 2005) se detallan los principales algoritmos de las ciencias de la computación para la gestión distribuida de procesos, los estados globales distribuidos y la exclusión mutua distribuida.

En (Tanenbaum, 1996 y 2009) se describen los principales algoritmos de sincronización en sistemas distribuidos (algoritmos clásicos de las ciencias de la computación).

En (La Red Martínez, 2017) se desarrollan operadores de agregación para asignaciones de recursos en sistemas distribuidos.

Sistemas de soporte de decisión

En (Chen, 2001) se estudia la aplicación de métodos lingüísticos de toma de decisiones para tratar el problemas de evaluación de calidad de servicio.

En (Doña et al., 2011) se presenta un modelo de decisión en grupo con la utilización de operadores de agregación de la familia OWA (Ordered Weighted Averaging: Promedio de Pesos Ordenados).

En (Fodor et al., 1994) se estudia el modelado de preferencias difuso para el soporte de decisiones multicriterio.

En (Fullér, 1996) se presenta la utilización de operadores de agregación de la familia OWA (Ordered Weighted Averaging: Promedio de Pesos Ordenados) para la toma de decisiones.

En (García-Melón et al., 2006) se presenta el ANP (Analytic Network Process: Proceso Analítico en Red) y su aplicación a un caso concreto de toma de decisiones. El ANP se basa en el MCDA (Multiple Criteria Decision Analysis: Análisis de Decisión Multi-Criterio).

En (Greco et al., 2002) se presentan metodologías para resolver problemas en presencia de múltiples atributos y criterios. En (Marakas, 2002) se estudian los sistemas de soporte de decisión.

En (Peláez et al., 2003a, 2003b, 2003, 2004, 2007, 2009) se analizan operadores de agregación (de mayoría) en grupo que buscan la representación de la mayoría.

En (Saaty, 1980) se presenta el AHP (Analytic Hierarchy Process: Proceso Analítico Jerárquico) para la toma de decisiones.

Toma de decisiones en grupo

En (Chao et al., 2016) se estudia la forma de obtener un vector de prioridades colectivo a partir de diferentes formatos de expresión de las preferencias por parte de los decisores. El modelo puede reducir la complejidad de la toma de decisiones y evitar la pérdida de información cuando se transforman los diferentes formatos en un formato único de expresión de las preferencias.

En (Chiclana et al., 2000), (Chiclana et al., 2001) y (Chiclana et al., 2004) se

presentan varios operadores de agregación utilizables para la toma de decisiones en grupos.

En (Dong et al., 2016b) se define un problema complejo y dinámico de toma de decisiones en grupo con múltiples atributos y se propone un método de resolución que utiliza un proceso de consenso para grupos de atributos, de alternativas y de preferencias, presentándose un modelo de decisión para problemas del mundo real.

En (La Red et al., 2011a) se presenta el operador *WKC-OWA* para agregar información en problemas de decisión democrática.

En (La Red y Acosta, 2015) se presentan las principales propiedades matemáticas y las medidas de comportamiento relacionadas con los operadores de agregación.

En (La Red Martínez y Pinto, 2015) se presenta una revisión acerca de los operadores de agregación, especialmente los de la familia *OWA*.

En (Lu et al., 2007), (Martínez et al., 2007) y (Martínez et al., 2006) se estudia la toma de decisiones en grupo con multi-objetivos, el tratamiento de información heterogénea en procesos ingenieriles de evaluación y en sistemas de ingeniería.

En (Peláez et al., 2003a, 2003b, 2004, 2007) se analizan operadores de agregación de mayoría y sus posibles aplicaciones a la toma de decisiones en grupo.

En (Yager, 1988, 1993) se presentan y analizan los operadores *OWA* (Ordered Weighted Averaging: Promedio de Pesos Ordenados) aplicados a la toma de decisiones multicriterio.

En (Yager et al., 1997, 2002) se presentan los operadores *OWA* y sus aplicaciones en la toma de decisiones multi-agente.

1.5. Metodología

La investigación que se realizará será de tipo teórica en la etapa de desarrollo de los modelos de decisión. Una vez definidos teóricamente los modelos de decisión antes

mencionados, se procederá a la validación de los mismos comparando sus prestaciones con las de los modelos de las ciencias de la computación habitualmente utilizados en los sistemas operativos. Como consecuencia del análisis, podría ser necesario modificar los modelos de decisión propuestos, lo cual daría inicio a un nuevo ciclo de validaciones.

Estructuras de datos

El sistema de matrices de datos que se utilizará contemplará las siguientes premisas y estructuras de datos. Se trata de grupos de procesos distribuidos en nodos de procesos que acceden a recursos críticos compartidos en la modalidad de exclusión mutua distribuida, debiendo decidirse, ante la demanda de recursos por parte de los procesos, cuáles serán las prioridades para asignar los recursos a los procesos que los requieren (sólo intervendrán como alternativas de asignación a los procesos aquellos recursos disponibles, es decir, no asignados ya a determinados procesos):

- El permiso de acceso a los recursos compartidos propios de un nodo no dependerá sólo de si los nodos los están utilizando o no, sino del valor de agregación de las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas).

- Las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas) dependerá de la consideración del valor de variables que representen el estado de cada uno de los distintos nodos. Cada nodo deberá expresar sus prioridades para la asignación de los distintos recursos compartidos respecto de los requerimientos de recursos de cada proceso de cada grupo.

Nodos que alojan procesos: $1, \dots, n$.

Procesos alojados en cada uno de los n nodos: $1, \dots, p$.

Grupos de procesos distribuidos: $1, \dots, g$.

Tamaño de cada uno de los g grupos de procesos: $1, \dots, t$.

Recursos críticos compartidos en la modalidad de exclusión mutua distribuida

disponibles en cada uno de los n nodos: $1, \dots, r$.

Estados posibles de cada uno de los p procesos:

- Grupo al que pertenece el proceso (0 significa proceso independiente).
- Requiere sincronización con los demás procesos del grupo al que pertenece (los procesos del grupo deben estar activos en sus respectivos procesadores en un mismo lapso de tiempo), (0 significa que no requiere sincronización, 1 significa que sí la requiere).
- En espera de un recurso compartido con el grupo de procesos al que pertenece.
- En espera de un recurso no compartido con el grupo de procesos al que pertenece.
- En ejecución con permiso de acceso a un recurso compartido con el grupo de procesos al que pertenece.
- En ejecución sin permiso de acceso a un recurso compartido con el grupo de procesos al que pertenece.

- Inactivo.

Estado posible de cada uno de los n nodos:

- Número de procesos.
- Prioridades de los procesos.
- Uso de CPU.
- Uso de memoria principal.
- Uso de memoria virtual.
- Estado de cada uno de los r recursos críticos compartidos en la modalidad de exclusión mutua distribuida existentes en el nodo:
 - o Asignado a un proceso local.
 - o Asignado a un proceso remoto.

o Disponible.

- Predisposición (prioridad nodal) para otorgar el acceso a cada uno de los r recursos críticos compartidos (alternativas) en la modalidad de exclusión mutua distribuida (resultará de la consideración de las variables representativas del estado del nodo, para cada recurso crítico compartido existente). Se obtendrá una tupla por cada uno de los n nodos, cada tupla contendrá r valores (prioridades nodales) para compartir los recursos críticos.

Estado global del sistema:

- Número de grupos de procesos y tamaño (número t de procesos) de cada uno de los g grupos.

- Porcentajes de consenso requeridos para otorgar el acceso a cada uno de los r recursos críticos disponibles en cada uno de los n nodos.

- Predisposición (prioridad global) para otorgar el acceso a cada uno de los r recursos críticos compartidos (alternativas) en la modalidad de exclusión mutua distribuida (resultará de la agregación de las prioridades nodales para cada recurso crítico compartido existente (alternativas). Se obtendrá una tupla con r valores normalizados (prioridades globales) para compartir los recursos críticos.

- Decisión de acceso a los r recursos críticos en función de contrastar las prioridades globales normalizadas para compartir los mismos con los porcentajes de consenso requeridos para otorgar los respectivos accesos.

- El estado global del sistema deberá actualizarse reiteradamente mientras haya alguno o algunos de los p procesos que requieran acceder a alguno o algunos de los r recursos compartidos.

El sistema se auto regula reiteradamente en función del estado local de los n nodos y del estado global del sistema, produciéndose una actualización de los estados locales de los nodos como consecuencia de la evolución de sus respectivos procesos y de las

decisiones de acceso a los recursos críticos producidas teniendo en cuenta el estado global del sistema: el sistema distribuido en el que se ejecutan grupos de procesos que acceden a recursos críticos, se observa a sí mismo y produce decisiones de accesos a recursos que modifican el estado del sistema y lo reajustan reiterativamente.

Tratamiento y análisis de datos

Se han efectuado simulaciones con los modelos de decisión propuestos y los modelos de decisión computacionales clásicos, a los efectos de analizar el comportamiento de los mismos para las mismas condiciones de carga de trabajo y de consumo de recursos, dando lugar a un esquema iterativo de modificación de los modelos propuestos para intentar lograr un desempeño de los mismos al menos equivalente al de los modelos computacionales clásicos.

El escenario propuesto en este documento considera los recursos y procesos de los sistemas operativos distribuidos, aplicados al entorno de las telecomunicaciones, pero sin limitarse a ningún protocolo de comunicaciones específico, se trata de un esquema genérico. Se considera que la aplicación del método propuesto supondría un aumento del tráfico de información de control, pero el rendimiento global del sistema mejoraría al asignar recursos a los procesos de acuerdo con un esquema de toma de decisiones holístico y cognitivo que también garantiza la exclusión mutua en el acceso a los recursos compartidos.

1.6. Estructura de la tesis

Se ha presentado la problemática que motivó la realización de este trabajo de investigación, así como los principales antecedentes y los conceptos teóricos que constituyen su marco conceptual, y los principales aspectos de la metodología a utilizar, se indica a continuación los restantes capítulos en que se ha estructurado esta tesis.

Capítulo II - Operador de agregación definido para el escenario 1 (E1). Se ha:

- Estudiado el escenario general E1.

- Formalizado el operador de agregación para el escenario E1.
- Validado empíricamente el operador de agregación para el escenario E1.
- Definido y validado teóricamente el modelo global de decisión del escenario E1.
- Definido detalladamente las etapas, niveles o capas del modelo de decisión E1 y validado empíricamente.

- Evaluado y comparado los resultados del modelo de decisión del escenario E1.

Capítulo III - Operador de agregación definido para el escenario 2 (E2). Se ha:

- Estudiado el escenario E2.
- Formalizado el operador de agregación para el escenario E2.
- Validado empíricamente el operador de agregación para el escenario E2.
- Definido y validado teóricamente el modelo global de decisión del escenario E2.
- Definido detalladamente las etapas, niveles o capas del modelo de decisión E2 y validado empíricamente.

- Evaluado y comparado los resultados del modelo de decisión del escenario E2.

Capítulo IV - Operador de agregación definido para el escenario 3 (E3). Se ha:

- Estudiado el escenario E3.
- Formalizado el operador de agregación para el escenario E3.
- Validado empíricamente el operador de agregación para el escenario E3.
- Definido y validado teóricamente el modelo global de decisión del escenario E3.
- Definido detalladamente las etapas, niveles o capas del modelo de decisión E3 y validado empíricamente.

- Evaluado y comparado los resultados del modelo de decisión del escenario E3.

Capítulo V - Operador de agregación definido para el escenario 4 (E4). Se ha:

- Estudiado el escenario E4.
- Formalizado el operador de agregación para el escenario E4.
- Validado empíricamente el operador de agregación para el escenario E4.
- Definido y validado teóricamente el modelo global de decisión del escenario E4.
- Definido detalladamente las etapas, niveles o capas del modelo de decisión E4 y

validado empíricamente.

- Evaluado y comparado los resultados del modelo de decisión del escenario E4.

Capítulo VI - Modelo de decisión: Se ha desarrollado y explicado el modelo de decisión propuesto para poder aplicar cada uno de los operadores de agregación para cada escenario.

Capítulo VII - Conclusiones y futuras líneas de investigación: se ha comentado las principales conclusiones y se han indicado las posibles líneas futuras de investigación.

1.7. Discusiones y comentarios

Se ha hecho la descripción de antecedentes, el marco teórico, en el que se detallan los principales aportes al estado del conocimiento en las distintas áreas relacionadas con la presente tesis (cibernética, comunicación y sincronización en sistemas distribuidos, toma de decisiones en grupo, etc). Se han planteado las hipótesis, que se contemplen situaciones en las que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua con o sin constituir grupos de procesos, que requieran o no sincronización y que puedan tener o no exigencias estrictas de consenso para lograr el acceso.

Se han descrito las estructuras de datos, los elementos que intervienen dentro del sistema, los nodos, recursos, procesos, que tienen prioridades y cargas de trabajo y que en diferentes situaciones permiten que el sistema se auto regula reiteradamente en función del estado local de los n nodos y del estado global del sistema.

Y finalmente se ha planteado la estructura de la tesis, además de lo antes mencionado, continúa con cuatro capítulos referentes a los distintos escenarios que conllevan al desarrollo de varios operadores de agregación, un capítulo sobre el modelo de decisión propuesto y se finaliza con las conclusiones y líneas futuras.

El comienzo de lo establecido en la estructura de la tesis antes mencionada inicia en el siguiente capítulo.

Capítulo II – Operador de agregación definido para el escenario 1 (E1)

2.1. Introducción

Trabajos previos

En (La Red Martínez, 2017) se desarrollan operadores de agregación para la asignación de recursos en sistemas distribuidos.

2.2. Hipótesis y objetivos

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 requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y **sin exigencias estrictas de consenso** para lograr el acceso (no se requiere 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 puede ser interrumpida para asignar recursos a otro proceso).

2.3. Propuesta de solución

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 que consignan fuente de elaboración propia, basado en La Red Martínez (2017), se transcriben en el ejemplo de este escenario (E1).

Se trata de grupos de procesos distribuidos en nodos de procesos que acceden a recursos críticos compartidos en la modalidad de exclusión mutua distribuida, debiendo decidirse, ante la demanda de recursos por parte de los procesos, cuáles serán las prioridades para asignar los recursos a los procesos que los requieren (sólo intervendrán como alternativas de asignación a los procesos aquellos recursos

disponibles, es decir, no asignados aún a determinados procesos):

- El permiso de acceso a los recursos compartidos propios de un nodo no dependerá sólo de si los nodos los están utilizando o no, sino del valor de agregación de las preferencias (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas).
- Las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas) dependerá de la consideración del valor de variables que representen el estado de cada uno de los distintos nodos. Cada nodo deberá expresar sus prioridades para la asignación de los distintos recursos compartidos respecto de los requerimientos de recursos de cada proceso.

Nodos que alojan procesos: $1, \dots, n$. El conjunto de nodos se representa de la siguiente manera:

$$\text{nodos} = \{n_1, \dots, n_n\}$$

Procesos alojados en cada uno de los n nodos: $1, \dots, p$. El conjunto de procesos se representa de la siguiente manera:

procesos = $\{p_{ij}\}$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, p$ (n° máximo de procesos en cada nodo), lo que se puede expresar mediante la Tabla 1.

TABLA 1. Procesos en cada nodo

Nodos	Procesos			
1	p_{11}	p_{12}	p_{1p}
....
i	p_{i1}	p_{i2}	p_{ip}
....
n	p_{n1}	p_{n2}	p_{np}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Grupos de procesos distribuidos: $1, \dots, g$. El conjunto de grupos de procesos distribuidos se representa de la siguiente manera:

$$\text{grupos} = \{p_{ij}\}$$

con i indicando el nodo y j el proceso en dicho nodo.

Tamaño de cada uno de los g grupos de procesos. El n° de procesos en cada

grupo indica la cardinalidad del grupo y se representa de la siguiente manera:

$$card = \{card(g_i)\} \text{ con } i = 1, \dots, g \text{ indicando el grupo.}$$

Prioridad grupal de cada uno de los g grupos de procesos. Estas prioridades se pueden fijar según distintos criterios; en esta propuesta se considerará que es función de la cardinalidad de cada grupo y se representa de la siguiente manera:

$$prg = \{prg_i = card(g_i)\} \text{ con } i = 1, \dots, g \text{ indicando el grupo.}$$

Recursos compartidos en la modalidad de exclusión mutua distribuida disponibles en los n nodos: $1, \dots, r$, El conjunto de recursos se representa de la siguiente manera:

$recursos = \{r_{ij}\}$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, r$ (n° máximo de recursos en cada nodo), lo que se puede expresar mediante la Tabla 2.

TABLA 2. Recursos compartidos disponibles en cada nodo

Nodos	Recursos			
1	r_{11}	r_{12}	r_{1r}
....
i	r_{i1}	r_{i2}	r_{ir}
....
n	r_{n1}	r_{n2}	r_{nr}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Estos recursos compartidos disponibles alojados en distintos nodos del sistema distribuido podrán ser requeridos por los procesos (agrupados o independientes) en ejecución en los nodos; estas solicitudes de recursos por parte de los procesos se muestran en la Tabla 3:

TABLA 3. Recursos solicitados por los procesos

Recursos		Procesos			
r_{11}	p_{11}	p_{kl}	p_{np}
....
r_{ij}	p_{11}	p_{kl}	p_{np}
....
r_{nr}	p_{11}	p_{kl}	p_{np}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Estados posibles de cada uno de los p procesos:

- Proceso independiente.
- Proceso perteneciente a un grupo de procesos.

Estado posible de cada uno de los n nodos:

- Número de procesos.
- Prioridades de los procesos.
- Uso de CPU.
- Uso de memoria principal.
- Uso de memoria virtual.
- Memoria adicional necesaria para cada recurso solicitado por cada proceso (si se dispone del dato).
- Carga adicional de procesador estimada necesaria para cada recurso solicitado por cada proceso (si se dispone del dato).
- Carga adicional de entrada/salida estimada necesaria para cada recurso solicitado por cada proceso (si se dispone del dato).
- Estado de cada uno de los r recursos compartidos en la modalidad de exclusión mutua distribuida existentes en el nodo:
 - Asignado a un proceso local o remoto.
 - Disponible.
- Predisposición (prioridad nodal) para otorgar el acceso a cada uno de los r recursos compartidos en la modalidad de exclusión mutua distribuida (resultará de la consideración de las variables representativas del estado del nodo, de la prioridad de los procesos y de la carga computacional adicional que significaría asignar el recurso al proceso solicitante).
- Carga actual del nodo, que se podrá calcular como el promedio de los porcentajes de uso de CPU, memoria y entrada/salida en un momento dado (estos indicadores de carga podrán variar según los casos, pudiendo agregarse otros o

cambiarse algunos de los puestos como ejemplo); también habrán de definirse las categorías de carga actual, por ejemplo, Alta, Media y Baja, señalándose los rangos de valores para cada categoría.

2.4. Descripción del operador de agregación

El operador propuesto consta de las siguientes etapas:

- 1) Cálculo de la carga computacional actual de los nodos.
- 2) Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas.
- 3) 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).
- 4) 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.

A continuación, se describirá cada una de las etapas mencionadas.

Cálculo de la carga computacional actual de los nodos

Para obtener un indicador de la carga computacional actual de cada nodo se pueden adoptar distintos criterios; en esta propuesta los criterios serán el % de uso de la CPU, el % de uso de la memoria y el % de uso de operaciones de entrada / salida, como se verá en el ejemplo.

La carga computacional de cada nodo se calculará de la siguiente manera:

Establecimiento del n° de criterios para determinar la carga de los nodos:

$$\text{card}(\{\text{criterios}\}) = c$$

Establecimiento de los criterios que se aplicarán (podrán diferir de un nodo a otro):

$criterios = \{c_{ij}\}$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, c$ (n° máximo de criterios para cada nodo), lo que se puede expresar mediante la Tabla 4.

TABLA 4. *Criterios para medir la carga computacional en cada nodo*

Nodos	Criterios			
1	c_{11}	c_{12}	c_{1c}
....
i	c_{i1}	c_{i2}	c_{ic}
....
n	c_{n1}	c_{n2}	c_{nc}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Eventualmente todos los nodos podrían utilizar el mismo conjunto de criterios.

Cálculo de la carga computacional de cada nodo:

$$carga_i = (\text{valor}(c_{i1}) + \dots + \text{valor}(c_{ic})) / c \text{ con } i = 1, \dots, n$$

Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas

Para establecer las categorías de carga computacional actual de cada nodo se pueden adoptar distintos criterios; en esta propuesta las categorías serán: Alta (si la carga es mayor al 70%), Media (si la carga está entre el 40% y el 70% inclusive) y Baja (si la carga es menor al 40%), como se verá en el ejemplo.

Establecimiento del n° de categorías para determinar la carga de los nodos:

$$card(\{categorías\}) = a$$

Establecimiento de las categorías que se aplicarán (podrán diferir de un nodo a otro):

$categorías = \{cat_{ij}\}$ con $i = 1, \dots, n$ (n° de nodos en el sistema distribuido) y $j = 1, \dots, a$ (n° máximo de categorías para cada nodo), lo que se puede expresar mediante la Tabla 5.

TABLA 5. *Categorías para medir la carga computacional en cada nodo*

Nodos	Categorías			
1	cat_{11}	cat_{12}	cat_{1a}
....
i	cat_{i1}	cat_{i2}	cat_{ia}
....
n	cat_{n1}	cat_{n2}	cat_{na}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Eventualmente todos los nodos podrían utilizar el mismo conjunto de categorías.

Para establecer los vectores de pesos asociados a las categorías de carga computacional actual de cada nodo se pueden adoptar distintos criterios; en esta propuesta los criterios serán: N° de procesos en el nodo, % de uso de CPU, % de uso de memoria, % de uso de memoria virtual, prioridad del proceso (prioridad del proceso en el nodo donde se ejecuta), sobrecarga de memoria (memoria adicional que requerirá disponer el recurso solicitado, si el dato está disponible), sobrecarga de procesador (uso adicional de procesador que requerirá disponer el recurso solicitado, si el dato está disponible) y sobrecarga de entrada / salida (entrada / salida adicional que requerirá disponer el recurso solicitado, si el dato está disponible), como se verá en el ejemplo.

Establecimiento del n° de criterios para determinar la prioridad o preferencia que se otorgará en cada nodo según su carga a cada pedido de un recurso compartido hecho por cada proceso:

$$card(\{critpref\}) = e$$

Establecimiento de los criterios que se aplicarán (iguales para todos los nodos):

criterios para preferencias = $\{cp_{ij}\}$ con $i = 1, \dots, a$ (n° de categorías de carga computacional) y $j = 1, \dots, e$ (n° máximo de criterios), lo que se puede expresar mediante la Tabla 6.

TABLA 6. *Criterios para calcular la prioridad o preferencia de cada nodo*

Categorías		Criterios		
1	cp_{11}	cp_{12}	cp_{1e}
....
i	cp_{i1}	cp_{i2}	cp_{ie}
....
a	cp_{a1}	cp_{a2}	cp_{ae}

Nota: Criterios para calcular la prioridad o preferencia que cada nodo otorgará a cada requerimiento de cada proceso según la carga del nodo

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Eventualmente todos los nodos podrían utilizar distintos conjuntos de criterios aplicables a las distintas categorías de carga computacional; en esta propuesta y como se verá en el ejemplo, se utilizan los mismos criterios para todos los nodos.

Una vez determinadas las categorías para indicar la carga de los nodos y los criterios que se aplicarán para evaluar la prioridad a otorgar a cada requerimiento de recursos de cada proceso, se podrán establecer los valores correspondientes a los criterios constituyendo así los vectores de pesos para las distintas categorías de carga.

Establecimiento de los vectores de pesos que se aplicarán (iguales para todos los nodos):

$w_{ij} = \{w_{ij}\}$ con $i = 1, \dots, a$ (n° de categorías de carga computacional) y $j = 1, \dots, e$ (n° máximo de criterios), lo que se puede expresar mediante la Tabla 7.

TABLA 7. Pesos asignados a los criterios de cálculo de preferencias

Categorías		Pesos			
1	w_{11}	w_{12}	w_{1e}	
....	
i	w_{i1}	w_{i2}	w_{ie}	
....	
a	w_{a1}	w_{a2}	w_{ae}	

Nota: Pesos asignados a los criterios para calcular la prioridad o preferencia que cada nodo otorgará a cada requerimiento de cada proceso según la carga del nodo.

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

La asignación de pesos a los distintos criterios será función de estudios estadísticos previamente realizados acerca del sistema distribuido; habrá entonces una función de asignación de pesos a los criterios para constituir los vectores de pesos de cada categoría de carga:

$w_{ij} = norm(función(cp_{ij}))$ con $i = 1, \dots, a$ (n° de categorías) y $j = 1, \dots, e$ (n° de criterios); *norm* indica que los valores deben estar normalizados (en el intervalo de 0 a 1 inclusive) y con la restricción de que la sumatoria de los elementos de un vector de pesos debe dar 1:

$$\sum \{w_{ij}\} = 1 \text{ con } j = 1, \dots, e \text{ para cada } i \text{ constante.}$$

Esto significa que la sumatoria de los pesos asignados a los distintos criterios será 1 para cada una de las categorías, o lo que es lo mismo, que la suma de elementos del vector de pesos de cada categoría es 1.

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 y podría llamárselas

prioridades nodales)

Estas prioridades son calculadas en cada nodo para cada petición de recursos originada en cada proceso; el cálculo considera el vector de pesos correspondiente según la carga actual del nodo y el vector de los valores otorgados por el nodo según los criterios de evaluación de la petición. El rango de valores es entre 0 y 1, donde un valor cercano a 0 significa que el criterio relacionado aportará poco al cálculo de la prioridad de la petición en tanto que un valor cercano a 1 significa lo contrario. Esto permite que un nodo ante una petición de un recurso por parte de un proceso, podrá influir en la misma según su estado y el impacto o carga adicional que significaría asignar el recurso solicitado al proceso solicitante, por ejemplo, si acceder a la petición significa incrementar el uso de memoria y el nodo tiene poca memoria disponible, entonces podría asignar a dicho criterio un valor cercano a 0, a su vez, si el consumo adicional de procesador se considera bajo y el uso de CPU del nodo es reducido, entonces a dicho criterio se asignaría un valor cercano a 1.

Los vectores de valoraciones que se aplicarán para cada requerimiento de un recurso por parte de un proceso, según los criterios establecidos para la determinación de la prioridad que en cada caso y momento fijará el nodo en el cual se produce el requerimiento, 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 $m = 1, \dots, e$ (criterios de valoración de la prioridad del requerimiento), los que se pueden expresar mediante la Tabla 8.

TABLA 8. Valoraciones asignadas a los criterios para el cálculo de prioridades

Recursos - Procesos		Criterios			
$r_{11} p_{11}$	cp_1	cp_m	cp_e
....
$r_{ij} p_{kl}$	cp_1	cp_m	cp_e
....
$r_{nr} p_{np}$	cp_1	cp_m	cp_e

Nota: Valoraciones asignadas a los criterios para calcular la prioridad o preferencia que cada nodo otorgará a cada requerimiento de cada proceso según la carga del nodo
Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Resumiendo, la prioridad nodal (por ser calculada en el nodo en el que se produce la petición) 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:

$prioridad\ nodal\ (r_{ij}\ p_{kl}) = \sum w_{om} * cp_m$ con o indicando el vector de pesos según la carga del nodo, manteniendo los demás subíndices los significados explicados anteriormente.

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

En esta etapa se consideran las prioridades nodales calculadas en la etapa anterior para cada requerimiento de acceso a los recursos por parte de los procesos. A partir de estas prioridades nodales se deben calcular las prioridades globales o finales, es decir, con qué prioridad, o sea en qué orden, los recursos solicitados serán otorgados y a qué procesos se hará dicho otorgamiento. Los requerimientos que no puedan ser atendidos por resultar con bajas prioridades, serán nuevamente considerados en la siguiente iteración del método.

Para el cálculo de las prioridades finales se utiliza la Tabla 9, en la cual 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 distintos procesos para acceder a un determinado recurso.

TABLA 9. *Prioridades nodales de los procesos para acceder a cada recurso*

Recursos		Prioridades Nodales de los Procesos			
r_{11}	p_{11}	p_{kl}	p_{np}
....
r_{ij}	p_{11}	p_{kl}	p_{np}
....
r_{nr}	p_{11}	p_{kl}	p_{np}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Seguidamente corresponde calcular el vector de pesos finales que se utilizará en

el proceso final de agregación para determinar el orden o prioridad de acceso a los recursos.

pesos finales = $\{wf_{kl}\}$ con $k = 1, \dots, n$ (n° de nodos) y $l = 1, \dots, p$ (n° máximo de procesos por nodo), lo que se puede expresar mediante la Tabla 10, donde np es el número de procesos en el sistema y prg_i es la prioridad del grupo de procesos al que pertenece el proceso (explicada anteriormente).

TABLA 10. Pesos asignados a los procesos para el cálculo de prioridades

Procesos	Pesos Finales	
	Si integra un grupo de procesos	Si es independiente
p_{11}	$wf_{11}=(prg_i)/np$	$wf_{11}=1/np$
....
p_{kl}	$wf_{kl}=(prg_i)/np$	$wf_{kl}=1/np$
....
p_{np}	$wf_{np}=(prg_i)/np$	$wf_{np}=1/np$

Nota: Pesos asignados a los procesos para calcular la prioridad o preferencia final de acceso a los recursos.

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

El siguiente paso es normalizar los pesos recientemente obtenidos dividiendo cada uno por la sumatoria de todos ellos, lo cual se indica en la Tabla 11.

TABLA 11. Pesos finales normalizados asignados a los procesos

Procesos	Pesos Finales Normalizados
p_{11}	$nwf_{11} = wf_{11} / \sum wf_{kl}$
....
p_{kl}	$nwf_{kl} = wf_{kl} / \sum wf_{kl}$
....
p_{np}	$nwf_{np} = wf_{np} / \sum wf_{kl}$

Nota: Pesos finales normalizados asignados a los procesos para calcular la prioridad o preferencia final de acceso a los recursos.

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Es así como se obtiene un vector de pesos normalizados (en el intervalo de 0 a 1 inclusive) y con la restricción de que la sumatoria de los elementos del vector debe dar 1:

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

Las prioridades nodales indicadas en la Tabla 9 tomadas fila por fila, es decir, respecto de cada recurso, se multiplicarán escalarmente por el vector de pesos finales normalizados indicado en la Tabla 11 para obtener las prioridades globales finales de

acceso de cada proceso a cada recurso y de allí, el orden o prioridad con que se asignarán los recursos y a qué proceso se asignará cada uno de ellos; esto se indica a continuación.

prioridad final global $(r_{ij} p_{kl}) = nwf_{kl} * p_{kl}$ con r_{ij} indicando el recurso j del nodo i , p_{kl} el proceso l del nodo k y el producto la prioridad final global de dicho proceso para acceder al mencionado recurso. El mayor de estos productos hechos para los distintos procesos en relación al mismo recurso indicará cuál de los procesos tendrá acceso al recurso.

La sumatoria de todos estos productos en relación al mismo recurso indicará la prioridad que tendrá dicho recurso para ser asignado, en relación a los demás recursos que también tendrán que ser asignados. Esto constituye lo que se denominará Función de Asignación para Sistemas Distribuidos (**FASD**):

$$FASD(r_{ij}) = \sum nwf_{kl} * p_{kl} = \text{prioridad de asignación del recurso } r_{ij}.$$

Calculando la **FASD** para todos los recursos se obtendrá un vector y, ordenando sus elementos de mayor a menor se obtendrá el orden prioritario de asignación de los recursos. Además, como ya se ha indicado, el mayor de los productos $nwf_{kl} * p_{kl}$ respecto de cada recurso indicará el proceso al cual será asignado el recurso. Esto se indica en la Tabla 12.

TABLA 12. Orden o prioridad final de asignación de los recursos a procesos

Orden de asignación de los recursos	Proceso al que se asignará el recurso
1°: r_{ij} del Máximo(FASD (r_{ij}))	p_{kl} del Máximo($nwf_{kl} * p_{kl}$) para el r_{ij} seleccionado
2°: r_{ij} del Máximo(FASD (r_{ij})) para los r_{ij} no asignados	p_{kl} del Máximo($nwf_{kl} * p_{kl}$) para el r_{ij} seleccionado
....
último: r_{ij} no asignado	p_{kl} del Máximo($nwf_{kl} * p_{kl}$) para el r_{ij} seleccionado

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

2.5. Ejemplo

En esta sección se explicará detalladamente un ejemplo de aplicación del operador de agregación propuesto.

El sistema de procesamiento distribuido tiene tres nodos:

$$nodos = \{1, 2, 3\}$$

Los procesos que se ejecutan en los nodos son los siguientes: tres procesos en el nodo 1, cinco procesos en el nodo 2 y siete procesos en el nodo 3.

$procesos = \{p_{ij}\}$ con i indicando el nodo y j indicando el proceso, lo que se puede expresar mediante la Tabla 13.

TABLA 13. Procesos en cada nodo

Nodos	Procesos						
1	p_{11}	p_{12}	p_{13}				
2	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}		
3	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Varios procesos son independientes y otros constituyen grupos de procesos cooperativos; en este ejemplo se considerarán cuatro grupos, lo que se puede expresar mediante la Tabla 14.

TABLA 14. Procesos en cada grupo

Grupos	Procesos		
1	p_{11}	p_{25}	p_{37}
2	p_{12}	p_{21}	
3	p_{22}	p_{31}	
4	p_{13}	p_{23}	p_{34}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

El n° de procesos en cada grupo indica la cardinalidad del grupo y se representa de la siguiente manera:

$$card = \{card(g_i)\} = \{3, 2, 2, 3\}$$
 con i indicando el grupo.

La prioridad de los grupos de procesos se considerará que es la cardinalidad de cada grupo y se representa de la siguiente manera:

$$prg = \{prg_i = card(g_i)\} = \{3, 2, 2, 3\}$$
 con i indicando el grupo.

Los recursos compartidos disponibles en los nodos son los siguientes: tres recursos en el nodo 1, cuatro recursos en el nodo 2 y tres recursos en el nodo 3.

$recursos = \{r_{ij}\}$ con i indicando el nodo y j indicando el proceso, lo que se puede expresar mediante la Tabla 15.

Las solicitudes de recursos por parte de los procesos se muestran en la Tabla

16.

TABLA 15. Recursos compartidos disponibles en cada nodo

Nodos		Recursos		
1	r_{11}	r_{12}	r_{13}	
2	r_{21}	r_{22}	r_{23}	r_{24}
3	r_{31}	r_{32}	r_{33}	

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 16. Recursos solicitados por los procesos

Recursos	Procesos
r_{11}	$p_{11}, p_{12}, p_{13}, p_{24}, p_{32}, p_{33}, p_{36}, p_{37}$
r_{12}	$p_{11}, p_{12}, p_{13}, p_{21}, p_{23}, p_{24}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$
r_{13}	$p_{13}, p_{21}, p_{31}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}$
r_{21}	$p_{11}, p_{12}, p_{13}, p_{22}, p_{25}, p_{33}, p_{36}, p_{37}$
r_{22}	$p_{11}, p_{12}, p_{13}, p_{21}, p_{22}, p_{33}, p_{34}, p_{35}, p_{36}$
r_{23}	$p_{11}, p_{21}, p_{24}, p_{32}, p_{33}, p_{34}$
r_{24}	$p_{11}, p_{23}, p_{24}, p_{34}, p_{35}, p_{36}$
r_{31}	$p_{12}, p_{13}, p_{21}, p_{22}, p_{23}, p_{31}, p_{34}, p_{35}, p_{36}$
r_{32}	$p_{13}, p_{23}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$
r_{33}	$p_{12}, p_{13}, p_{21}, p_{22}, p_{23}, p_{31}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}$

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

A continuación, se describirá cada una de las etapas de cálculo.

Cálculo de la carga computacional actual de los nodos

Para obtener un indicador de la carga computacional actual de cada nodo se adoptarán los mismos tres criterios en los tres nodos:

$$card(\{criterios\}) = 3$$

$criterios = \{\% \text{ de uso de la CPU, el \% de uso de la memoria, \% de uso de operaciones de entrada / salida}\}$.

Los valores que se asumirán para los indicadores de carga computacional de los tres nodos y el cálculo de carga promedio para cada nodo se muestran en la Tabla 17.

TABLA 17. Valores de los criterios de carga computacional

Nodos	Valores de los Criterios			
1	80	90	75	Promedio: 81.67
2	45	50	65	Promedio: 53.33
3	10	25	35	Promedio: 23.33

Nota: Valores de los criterios para medir la carga computacional en cada nodo.

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas.

En esta propuesta las categorías serán las mismas para todos los nodos: Alta (si

la carga es mayor al 70%), Media (si la carga está entre el 40% y el 70% inclusive) y Baja (si la carga es menor al 40%).

$$\text{card}(\{\text{categorías}\}) = 3$$

$$\text{categorías} = \{\text{Alta, Media, Baja}\}.$$

Los valores que se obtienen para las categorías de carga en base a los promedios mostrados en la Tabla 17, se indican en la Tabla 18.

TABLA 18. Valores de las categorías para medir la carga computacional

Nodos	Valores de las Categorías
1	Alta
2	Media
3	Baja

Nota: Valores de las categorías para medir la carga computacional en cada nodo.
Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Para establecer los vectores de pesos asociados a las categorías de carga computacional actual de cada nodo se utilizarán, para todos los nodos y para todas las categorías de carga, los siguientes criterios: N° de procesos en el nodo, % de uso de CPU, % de uso de memoria, % de uso de memoria virtual, prioridad del proceso (prioridad del proceso en el nodo donde se ejecuta), sobrecarga de memoria (memoria adicional que requerirá disponer el recurso solicitado, si el dato está disponible), sobrecarga de procesador (uso adicional de procesador que requerirá disponer el recurso solicitado, si el dato está disponible) y sobrecarga de entrada / salida (entrada / salida adicional que requerirá disponer el recurso solicitado, si el dato está disponible).

$$\text{card}(\{\text{critpref}\}) = 8$$

criterios para preferencias = {N° de procesos en el nodo, % de uso de CPU, % de uso de memoria, % de uso de memoria virtual, prioridad del proceso, sobrecarga de memoria, sobrecarga de procesador, sobrecarga de entrada / salida}

Seguidamente se deben establecer los valores correspondientes a los criterios constituyendo así los vectores de pesos para las distintas categorías de carga, que serán iguales para todos los nodos, lo cual se indica en la Tabla 19.

La sumatoria de los pesos asignados a los distintos criterios es 1 para cada una de las categorías, o lo que es lo mismo, que la suma de elementos del vector de pesos de cada categoría es 1.

TABLA 19. Pesos asignados a los criterios para calcular la prioridad

Categorías	Pesos							
	N° Proc.	% CPU	% Mem.	% MV	Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S
Alta	0.050	0.050	0.100	0.500	0.100	0.100	0.050	0.050
Media	0.100	0.200	0.300	0.100	0.200	0.050	0.025	0.025
Baja	0.100	0.300	0.200	0.200	0.100	0.025	0.025	0.050

Nota: Pesos asignados a los criterios para calcular la prioridad o preferencia que cada nodo otorgará a cada requerimiento de cada proceso según la carga del nodo.

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

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 y podría llamarlas prioridades nodales)

Los vectores de valoraciones se aplican para cada requerimiento de un recurso hecho por un proceso, según los criterios establecidos para la determinación de la prioridad que en cada caso y momento fija el nodo en el cual se produce el requerimiento; cada vector de valoraciones de cada requerimiento se multiplica escalaramente por el vector de pesos correspondiente a la categoría de carga actual del nodo para obtener la prioridad según cada criterio y la prioridad nodal otorgada a cada requerimiento; esto se muestra en la Tabla 20.

TABLA 20. Valoraciones asignadas a los criterios para el cálculo de prioridad

Procesos - Recursos	Criterios								Prioridades Nodales
	N° Proc.	% CPU	% Mem.	% MV	Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S	
p_{11r11}	0.7000	0.5000	0.7000	0.9000	0.8000	0.2000	0.3000	0.4000	
$Pri(p_{11r11})$	0.0350	0.0250	0.0700	0.4500	0.0800	0.0200	0.0150	0.0200	0.7150
p_{11r12}	0.8000	0.7000	0.4000	0.5000	0.3000	0.7000	0.2000	0.4000	
$Pri(p_{11r12})$	0.0400	0.0350	0.0400	0.2500	0.0300	0.0700	0.0100	0.0200	0.4950
p_{11r21}	0.3000	0.4000	0.5000	0.2000	0.9000	0.2000	0.5000	0.7000	
$Pri(p_{11r21})$	0.0150	0.0200	0.0500	0.1000	0.0900	0.0200	0.0250	0.0350	0.3550
p_{11r22}	0.5000	0.5000	0.7000	0.4000	0.8000	0.3000	0.5000	0.6000	
$Pri(p_{11r22})$	0.0250	0.0250	0.0700	0.2000	0.0800	0.0300	0.0250	0.0300	0.4850
p_{11r23}	0.5000	0.6000	0.8000	0.8000	0.9500	0.9000	0.7000	0.6000	
$Pri(p_{11r23})$	0.0250	0.0300	0.0800	0.4000	0.0950	0.0900	0.0350	0.0300	0.7850
p_{11r24}	0.3000	0.5000	0.9000	0.2000	0.6000	0.6000	0.7000	0.4000	
$Pri(p_{11r24})$	0.0150	0.0250	0.0900	0.1000	0.0600	0.0600	0.0350	0.0200	0.4050
p_{12r11}	0.4000	0.7000	0.5000	0.9000	1.0000	0.9000	0.8000	0.8000	
$Pri(p_{12r11})$	0.0200	0.0350	0.0500	0.4500	0.1000	0.0900	0.0400	0.0400	0.8250
p_{12r12}	0.2000	0.7000	0.3000	0.7000	0.8000	0.3000	0.8000	0.9000	
$Pri(p_{12r12})$	0.0100	0.0350	0.0300	0.3500	0.0800	0.0300	0.0400	0.0450	0.6200
p_{12r21}	0.7000	0.4000	0.3000	0.7000	0.8000	0.9000	0.5000	0.2000	
$Pri(p_{12r21})$	0.0350	0.0200	0.0300	0.3500	0.0800	0.0900	0.0250	0.0100	0.6400
p_{12r22}	0.9000	0.6000	0.7000	0.7000	0.8000	0.2000	0.5000	0.4000	

TABLA 20. Valoraciones asignadas a los criterios para el cálculo de prioridad

Procesos	Criterios								Prioridades Nodales
	Nº Recursos	% Proc.	% CPU	% Mem.	% MV	Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	
Pri(p _{12f22})	0.0450	0.0300	0.0700	0.3500	0.0800	0.0200	0.0250	0.0200	0.6400
p _{12f31}	0.2000	0.5000	0.7000	0.7000	0.3000	0.2000	0.7000	0.8000	
Pri(p _{12f31})	0.0100	0.0250	0.0700	0.3500	0.0300	0.0200	0.0350	0.0400	0.5800
p _{12f33}	0.4000	0.5000	0.7000	0.9000	0.3000	0.4000	0.5000	0.8000	
Pri(p _{12f33})	0.0200	0.0250	0.0700	0.4500	0.0300	0.0400	0.0250	0.0400	0.7000
p _{13f11}	0.5000	0.7000	0.7000	0.8000	0.6000	0.5000	0.7000	0.8000	
Pri(p _{13f11})	0.0250	0.0350	0.0700	0.4000	0.0600	0.0500	0.0350	0.0400	0.7150
p _{13f12}	0.7000	0.8000	0.7000	0.4000	0.9000	0.2000	0.9000	0.7000	
Pri(p _{13f12})	0.0350	0.0400	0.0700	0.2000	0.0900	0.0200	0.0450	0.0350	0.5350
p _{13f13}	0.7000	0.6000	0.7000	0.8000	0.9000	0.4000	0.8000	0.7000	
Pri(p _{13f13})	0.0350	0.0300	0.0700	0.4000	0.0900	0.0400	0.0400	0.0350	0.7400
p _{13f21}	0.7000	0.4000	0.9000	0.3000	0.5000	0.7000	0.2000	0.3000	
Pri(p _{13f21})	0.0350	0.0200	0.0900	0.1500	0.0500	0.0700	0.0100	0.0150	0.4400
p _{13f22}	0.5000	0.9000	0.8000	0.3000	0.5000	0.4000	0.9000	0.3000	
Pri(p _{13f22})	0.0250	0.0450	0.0800	0.1500	0.0500	0.0400	0.0450	0.0150	0.4500
p _{13f31}	0.5000	0.7000	0.3000	0.6000	0.8000	0.9000	0.9000	0.5000	
Pri(p _{13f31})	0.0250	0.0350	0.0300	0.3000	0.0800	0.0900	0.0450	0.0250	0.6300
p _{13f32}	0.6000	0.9000	0.3000	0.6000	0.4000	0.8000	0.7000	0.8000	
Pri(p _{13f32})	0.0300	0.0450	0.0300	0.3000	0.0400	0.0800	0.0350	0.0400	0.6000
p _{13f33}	0.6000	0.2000	0.4000	0.6000	0.9000	0.8000	0.5000	0.8000	
Pri(p _{13f33})	0.0300	0.0100	0.0400	0.3000	0.0900	0.0800	0.0250	0.0400	0.6150
p _{21f12}	0.2000	0.1000	0.3000	0.8000	0.7000	0.7000	0.5000	0.8000	
Pri(p _{21f12})	0.0200	0.0200	0.0900	0.0800	0.1400	0.0350	0.0125	0.0200	0.4175
p _{21f13}	0.2000	0.4000	0.8000	0.9000	0.7000	0.4000	0.5000	0.2000	
Pri(p _{21f13})	0.0200	0.0800	0.2400	0.0900	0.1400	0.0200	0.0125	0.0050	0.6075
p _{21f22}	0.3000	0.5000	0.8000	0.9000	0.5000	0.6000	0.2000	0.4000	
Pri(p _{21f22})	0.0300	0.1000	0.2400	0.0900	0.1000	0.0300	0.0050	0.0100	0.6050
p _{21f23}	0.7000	0.5000	0.6000	0.8000	0.5000	0.2000	0.9000	0.4000	
Pri(p _{21f23})	0.0700	0.1000	0.1800	0.0800	0.1000	0.0100	0.0225	0.0100	0.5725
p _{21f31}	0.7000	0.5000	0.8000	0.6000	0.9000	0.4000	0.9000	0.9000	
Pri(p _{21f31})	0.0700	0.1000	0.2400	0.0600	0.1800	0.0200	0.0225	0.0225	0.7150
p _{21f33}	0.7000	0.4000	0.9000	0.8000	0.4000	0.8000	0.6000	0.6000	
Pri(p _{21f33})	0.0700	0.0800	0.2700	0.0800	0.0800	0.0400	0.0150	0.0150	0.6500
p _{22f21}	0.5000	0.4000	0.7000	0.8000	0.6000	0.4000	0.6000	0.9000	
Pri(p _{22f21})	0.0500	0.0800	0.2100	0.0800	0.1200	0.0200	0.0150	0.0225	0.5975
p _{22f22}	0.5000	0.9000	0.6000	0.8000	0.9000	0.4000	0.2000	0.1000	
Pri(p _{22f22})	0.0500	0.1800	0.1800	0.0800	0.1800	0.0200	0.0050	0.0025	0.6975
p _{22f31}	0.5000	0.4000	0.5000	0.2000	0.4000	0.8000	0.2000	0.9000	
Pri(p _{22f31})	0.0500	0.0800	0.1500	0.0200	0.0800	0.0400	0.0050	0.0225	0.4475
p _{22f33}	0.8000	0.4000	0.3000	0.2000	0.8000	0.9000	0.5000	0.8000	
Pri(p _{22f33})	0.0800	0.0800	0.0900	0.0200	0.1600	0.0450	0.0125	0.0200	0.5075
p _{23f12}	0.5000	0.6000	0.8000	0.3000	0.5000	0.7000	0.5000	0.5000	
Pri(p _{23f12})	0.0500	0.1200	0.2400	0.0300	0.1000	0.0350	0.0125	0.0125	0.6000
p _{23f24}	0.6000	0.2000	0.1000	0.7000	0.3000	0.8000	0.9000	0.4000	
Pri(p _{23f24})	0.0600	0.0400	0.0300	0.0700	0.0600	0.0400	0.0225	0.0100	0.3325
p _{23f31}	0.3000	0.1000	0.4000	0.8000	0.7000	0.9000	0.4000	0.6000	
Pri(p _{23f31})	0.0300	0.0200	0.1200	0.0800	0.1400	0.0450	0.0100	0.0150	0.4600
p _{23f32}	0.4000	0.4000	0.8000	0.2000	0.4000	0.6000	0.6000	0.6000	
Pri(p _{23f32})	0.0400	0.0800	0.2400	0.0200	0.0800	0.0300	0.0150	0.0150	0.5200
p _{23f33}	0.3000	0.6000	0.8000	0.2000	0.9000	0.6000	0.4000	0.2000	
Pri(p _{23f33})	0.0300	0.1200	0.2400	0.0200	0.1800	0.0300	0.0100	0.0050	0.6350
p _{24f11}	0.4000	0.6000	0.7000	0.9000	0.5000	0.7000	0.9000	0.5000	
Pri(p _{24f11})	0.0400	0.1200	0.2100	0.0900	0.1000	0.0350	0.0225	0.0125	0.6300
p _{24f12}	0.3000	0.6000	0.7000	0.8000	0.9000	0.7000	0.6000	0.9000	
Pri(p _{24f12})	0.0300	0.1200	0.2100	0.0800	0.1800	0.0350	0.0150	0.0225	0.6925
p _{24f23}	0.4000	0.9000	0.4000	0.8000	0.8000	0.7000	0.6000	0.3000	
Pri(p _{24f23})	0.0400	0.1800	0.1200	0.0800	0.1600	0.0350	0.0150	0.0075	0.6375
p _{24f24}	0.5000	0.8000	0.7000	0.9000	0.3000	0.4000	0.8000	0.7000	
Pri(p _{24f24})	0.0500	0.1600	0.2100	0.0900	0.0600	0.0200	0.0200	0.0175	0.6275
p _{25f21}	0.2000	0.8000	0.8000	0.9000	0.4000	0.5000	0.6000	0.8000	
Pri(p _{25f21})	0.0200	0.1600	0.2400	0.0900	0.0800	0.0250	0.0150	0.0200	0.6500
p _{31f13}	0.6000	0.9000	0.6000	0.9000	0.7000	0.4000	0.9000	0.8000	

TABLA 20. Valoraciones asignadas a los criterios para el cálculo de prioridad

Procesos	Criterios								Prioridades Nodales
	Nº Recursos	% Proc.	% CPU	% Mem.	% MV	Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	
Pri(p _{31r13})	0.0600	0.2700	0.1200	0.1800	0.0700	0.0100	0.0225	0.0400	0.7725
p _{31r31}	0.8000	0.3000	0.9000	0.5000	0.7000	0.3000	0.9000	0.7000	
Pri(p _{31r31})	0.0800	0.0900	0.1800	0.1000	0.0700	0.0075	0.0225	0.0350	0.5850
p _{31r33}	0.4000	0.7000	0.7000	0.5000	0.9000	0.9000	0.7000	0.7000	
Pri(p _{31r33})	0.0400	0.2100	0.1400	0.1000	0.0900	0.0225	0.0175	0.0350	0.6550
p _{32r11}	0.6000	0.9000	0.8000	0.5000	0.9000	0.7000	0.3000	0.7000	
Pri(p _{32r11})	0.0600	0.2700	0.1600	0.1000	0.0900	0.0175	0.0075	0.0350	0.7400
p _{32r12}	0.7000	0.4000	0.6000	0.9000	0.8000	1.0000	0.9000	0.7000	
Pri(p _{32r12})	0.0700	0.1200	0.1200	0.1800	0.0800	0.0250	0.0225	0.0350	0.6525
p _{32r13}	0.8000	0.9000	1.0000	0.7000	0.9000	0.3000	0.5000	0.9000	
Pri(p _{32r13})	0.0800	0.2700	0.2000	0.1400	0.0900	0.0075	0.0125	0.0450	0.8450
p _{32r23}	0.8000	0.8000	1.0000	0.9000	0.6000	0.8000	0.4000	0.9000	
Pri(p _{32r23})	0.0800	0.2400	0.2000	0.1800	0.0600	0.0200	0.0100	0.0450	0.8350
p _{33r11}	0.2000	0.7000	0.9000	0.8000	0.6000	0.9000	1.0000	0.3000	
Pri(p _{33r11})	0.0200	0.2100	0.1800	0.1600	0.0600	0.0225	0.0250	0.0150	0.6925
p _{33r12}	0.9000	1.0000	0.9000	0.7000	0.3000	0.5000	0.7000	0.3000	
Pri(p _{33r12})	0.0900	0.3000	0.1800	0.1400	0.0300	0.0125	0.0175	0.0150	0.7850
p _{33r13}	0.9000	0.5000	0.7000	0.9000	0.3000	0.4000	0.5000	0.8000	
Pri(p _{33r13})	0.0900	0.1500	0.1400	0.1800	0.0300	0.0100	0.0125	0.0400	0.6525
p _{33r21}	0.4000	0.6000	0.7000	0.8000	0.8000	0.4000	0.8000	0.8000	
Pri(p _{33r21})	0.0400	0.1800	0.1400	0.1600	0.0800	0.0100	0.0200	0.0400	0.6700
p _{33r22}	0.9000	0.4000	0.7000	0.8000	0.7000	0.4000	0.9000	0.7000	
Pri(p _{33r22})	0.0900	0.1200	0.1400	0.1600	0.0700	0.0100	0.0225	0.0350	0.6475
p _{33r23}	0.4000	0.6000	0.9000	1.0000	0.6000	0.4000	0.7000	0.8000	
Pri(p _{33r23})	0.0400	0.1800	0.1800	0.2000	0.0600	0.0100	0.0175	0.0400	0.7275
p _{33r32}	0.6000	0.6000	0.7000	0.8000	0.4000	0.5000	0.8000	0.8000	
Pri(p _{33r32})	0.0600	0.1800	0.1400	0.1600	0.0400	0.0125	0.0200	0.0400	0.6525
p _{33r33}	0.6000	1.0000	0.7000	0.4000	0.6000	0.8000	0.8000	0.9000	
Pri(p _{33r33})	0.0600	0.3000	0.1400	0.0800	0.0600	0.0200	0.0200	0.0450	0.7250
p _{34r12}	0.8000	1.0000	0.3000	0.5000	0.7000	0.3000	0.8000	0.7000	
Pri(p _{34r12})	0.0800	0.3000	0.0600	0.1000	0.0700	0.0075	0.0200	0.0350	0.6725
p _{34r13}	0.2000	1.0000	0.8000	0.5000	0.8000	0.6000	0.9000	0.7000	
Pri(p _{34r13})	0.0200	0.3000	0.1600	0.1000	0.0800	0.0150	0.0225	0.0350	0.7325
p _{34r22}	0.9000	0.8000	0.6000	0.8000	0.9000	0.8000	0.5000	0.6000	
Pri(p _{34r22})	0.0900	0.2400	0.1200	0.1600	0.0900	0.0200	0.0125	0.0300	0.7625
p _{34r23}	0.4000	0.6000	0.7000	0.8000	0.9000	0.5000	0.9000	0.6000	
Pri(p _{34r23})	0.0400	0.1800	0.1400	0.1600	0.0900	0.0125	0.0225	0.0300	0.6750
p _{34r24}	0.9000	0.4000	0.7000	0.4000	0.7000	0.5000	0.9000	0.7000	
Pri(p _{34r24})	0.0900	0.1200	0.1400	0.0800	0.0700	0.0125	0.0225	0.0350	0.5700
p _{34r31}	0.5000	0.6000	0.7000	0.9000	0.7000	0.9000	0.6000	0.7000	
Pri(p _{34r31})	0.0500	0.1800	0.1400	0.1800	0.0700	0.0225	0.0150	0.0350	0.6925
p _{34r32}	0.8000	0.6000	0.9000	0.5000	0.6000	0.9000	0.3000	0.9000	
Pri(p _{34r32})	0.0800	0.1800	0.1800	0.1000	0.0600	0.0225	0.0075	0.0450	0.6750
p _{34r33}	0.4000	0.6000	0.8000	0.5000	0.9000	0.9000	0.4000	0.3000	
Pri(p _{34r33})	0.0400	0.1800	0.1600	0.1000	0.0900	0.0225	0.0100	0.0150	0.6175
p _{35r12}	0.2000	0.4000	0.7000	0.4000	0.9000	0.5000	0.7000	0.9000	
Pri(p _{35r12})	0.0200	0.1200	0.1400	0.0800	0.0900	0.0125	0.0175	0.0450	0.5250
p _{35r13}	0.8000	0.9000	0.7000	0.3000	0.9000	0.8000	0.7000	0.4000	
Pri(p _{35r13})	0.0800	0.2700	0.1400	0.0600	0.0900	0.0200	0.0175	0.0200	0.6975
p _{35r22}	0.3000	0.8000	0.9000	0.4000	0.8000	0.6000	0.3000	0.6000	
Pri(p _{35r22})	0.0300	0.2400	0.1800	0.0800	0.0800	0.0150	0.0075	0.0300	0.6625
p _{35r24}	0.5000	0.8000	0.9000	0.4000	0.6000	0.9000	0.4000	0.7000	
Pri(p _{35r24})	0.0500	0.2400	0.1800	0.0800	0.0600	0.0225	0.0100	0.0350	0.6775
p _{35r31}	0.9000	0.8000	0.7000	0.4000	0.8000	0.3000	0.4000	0.8000	
Pri(p _{35r31})	0.0900	0.2400	0.1400	0.0800	0.0800	0.0075	0.0100	0.0400	0.6875
p _{35r32}	0.9000	0.7000	0.8000	0.6000	0.4000	0.9000	0.4000	0.7000	
Pri(p _{35r32})	0.0900	0.2100	0.1600	0.1200	0.0400	0.0225	0.0100	0.0350	0.6875
p _{35r33}	0.5000	0.7000	0.6000	0.9000	0.8000	0.5000	0.9000	0.7000	
Pri(p _{35r33})	0.0500	0.2100	0.1200	0.1800	0.0800	0.0125	0.0225	0.0350	0.7100
p _{36r11}	0.8000	0.9000	0.6000	0.5000	0.8000	0.9000	0.9000	0.6000	
Pri(p _{36r11})	0.0800	0.2700	0.1200	0.1000	0.0800	0.0225	0.0225	0.0300	0.7250
p _{36r12}	0.7000	0.9000	0.4000	0.9000	0.8000	0.9000	0.4000	0.7000	

TABLA 20. Valoraciones asignadas a los criterios para el cálculo de prioridad

Procesos - Recursos	Criterios								Prioridades Nodales
	N° Proc.	% CPU	% Mem.	% MV	Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S	
<i>Pri(p_{36r12})</i>	0.0700	0.2700	0.0800	0.1800	0.0800	0.0225	0.0100	0.0350	0.7475
<i>p_{36r13}</i>	0.9000	0.9000	0.8000	0.7000	0.8000	0.7000	0.9000	0.7000	
<i>Pri(p_{36r13})</i>	0.0900	0.2700	0.1600	0.1400	0.0800	0.0175	0.0225	0.0350	0.8150
<i>p_{36r21}</i>	0.8000	0.9000	0.5000	0.3000	0.7000	0.7000	0.9000	0.9000	
<i>Pri(p_{36r21})</i>	0.0800	0.2700	0.1000	0.0600	0.0700	0.0175	0.0225	0.0450	0.6650
<i>p_{36r22}</i>	0.8000	0.2000	0.1000	0.3000	0.8000	0.7000	0.6000	0.9000	
<i>Pri(p_{36r22})</i>	0.0800	0.0600	0.0200	0.0600	0.0800	0.0175	0.0150	0.0450	0.3775
<i>p_{36r24}</i>	0.4000	0.7000	0.9000	0.3000	0.8000	0.5000	0.8000	0.9000	
<i>Pri(p_{36r24})</i>	0.0400	0.2100	0.1800	0.0600	0.0800	0.0125	0.0200	0.0450	0.6475
<i>p_{36r31}</i>	0.5000	0.9000	0.9000	0.7000	0.8000	0.8000	0.8000	0.7000	
<i>Pri(p_{36r31})</i>	0.0500	0.2700	0.1800	0.1400	0.0800	0.0200	0.0200	0.0350	0.7950
<i>p_{36r32}</i>	0.9000	0.8000	0.6000	0.7000	0.8000	0.5000	0.6000	0.7000	
<i>Pri(p_{36r32})</i>	0.0900	0.2400	0.1200	0.1400	0.0800	0.0125	0.0150	0.0350	0.7325
<i>p_{36r33}</i>	0.2000	0.7000	0.4000	0.8000	0.8000	0.9000	0.4000	0.5000	
<i>Pri(p_{36r33})</i>	0.0200	0.2100	0.0800	0.1600	0.0800	0.0225	0.0100	0.0250	0.6075
<i>p_{37r11}</i>	0.9000	0.6000	0.8000	0.6000	0.9000	0.7000	0.9000	0.8000	
<i>Pri(p_{37r11})</i>	0.0900	0.1800	0.1600	0.1200	0.0900	0.0175	0.0225	0.0400	0.7200
<i>p_{37r12}</i>	0.7000	0.9000	0.8000	0.7000	0.5000	0.8000	0.8000	0.6000	
<i>Pri(p_{37r12})</i>	0.0700	0.2700	0.1600	0.1400	0.0500	0.0200	0.0200	0.0300	0.7600
<i>p_{37r21}</i>	0.9000	0.7000	0.8000	0.6000	0.6000	0.9000	0.5000	0.4000	
<i>Pri(p_{37r21})</i>	0.0900	0.2100	0.1600	0.1200	0.0600	0.0225	0.0125	0.0200	0.6950
<i>p_{37r32}</i>	0.8000	0.9000	0.5000	0.3000	0.8000	0.7000	0.4000	0.6000	
<i>Pri(p_{37r32})</i>	0.0800	0.2700	0.1000	0.0600	0.0800	0.0175	0.0100	0.0300	0.6475
<i>p_{37r33}</i>	0.8000	0.4000	0.6000	0.8000	0.8000	0.6000	0.9000	0.3000	
<i>Pri(p_{37r33})</i>	0.0800	0.1200	0.1200	0.1600	0.0800	0.0150	0.0225	0.0150	0.6125

Nota: Valoraciones asignadas a los criterios para calcular la prioridad o preferencia nodal que cada nodo otorgará a cada requerimiento de cada proceso según la carga del nodo.
Fuente: Elaboración propia. Basado en La Red Martínez (2017).

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

A partir de las prioridades nodales se deben calcular las prioridades globales o finales, es decir, con qué prioridad, o sea en qué orden, los recursos solicitados serán otorgados y a qué procesos se hará dicho otorgamiento. Para el cálculo de las prioridades finales se utiliza la Tabla 23.

Seguidamente corresponde calcular el vector de pesos finales que se utilizará en el proceso final de agregación para determinar el orden o prioridad de acceso a los recursos; esto se muestra en la Tabla 24.

Las prioridades nodales indicadas en la Tabla 23 tomadas fila por fila, es decir, respecto de cada recurso, se multiplicarán escalarmente por el vector de pesos finales

normalizados indicado en la Tabla 24 para obtener las prioridades globales finales de acceso de cada proceso a cada recurso y de allí, el orden o prioridad con que se asignarán los recursos y a qué proceso se asignará cada uno de ellos; esto se indica en la Tabla 25.

El mayor de estos productos hechos para los distintos procesos en relación al mismo recurso indicará cuál de los procesos tendrá acceso al recurso (en caso de empates se podría optar por dar ganador al proceso identificado con el número menor); esto se muestra en rojo en la Tabla 25.

La sumatoria de todos estos productos en relación al mismo recurso indicará la prioridad que tendrá dicho recurso para ser asignado. Esto constituye la Función de Asignación para Sistemas Distribuidos (**FASD**), que se muestra en la Tabla 21.

TABLA 21. Prioridades globales finales para asignar los recursos

FASD	Prioridad Global Final para Asignar el Recurso	Proceso
r_{11}	0.35120968	r_{11} al p_{37}
r_{12}	0.47306452	r_{12} al p_{37}
r_{13}	0.32862903	r_{13} al p_{13}
r_{21}	0.33000000	r_{21} al p_{37}
r_{22}	0.34403226	r_{22} al p_{34}
r_{23}	0.24919355	r_{23} al p_{11}
r_{24}	0.18951613	r_{24} al p_{34}
r_{31}	0.37048387	r_{31} al p_{34}
r_{32}	0.30322581	r_{32} al p_{34}
r_{33}	0.46798387	r_{33} al p_{23}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

El orden final de asignación de los recursos y los procesos destinatarios de los mismos se obtiene ordenando la Tabla 21 (**FASD**), lo cual se muestra en la Tabla 22.

TABLA 22. Orden o prioridad final de asignación

Prioridad Global Final Ordenada	Asignación
0.47306452	r_{12} al p_{37}
0.46798387	r_{33} al p_{23}
0.37048387	r_{31} al p_{34}
0.35120968	r_{11} al p_{37}
0.34403226	r_{22} al p_{34}
0.33000000	r_{21} al p_{37}
0.32862903	r_{13} al p_{13}
0.30322581	r_{32} al p_{34}
0.24919355	r_{23} al p_{11}
0.18951613	r_{24} al p_{34}

Nota: Orden o prioridad final de asignación de los recursos y proceso al cual se asigna cada recurso **FASDO**.

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Esto constituye la Función de Asignación para Sistemas Distribuidos Ordenada (**FASDO**).

TABLA 23. Prioridades nodales de los procesos para acceder a cada recurso

Recursos	Prioridades Nodales de los Procesos														
	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}
r_{11}	0.7150	0.8250	0.7150	0.0000	0.0000	0.0000	0.6300	0.0000	0.0000	0.7400	0.6925	0.0000	0.0000	0.7250	0.7200
r_{12}	0.4950	0.6200	0.5350	0.4175	0.0000	0.6000	0.6925	0.0000	0.0000	0.6525	0.7850	0.6725	0.5250	0.7475	0.7600
r_{13}	0.0000	0.0000	0.7400	0.6075	0.0000	0.0000	0.0000	0.0000	0.7725	0.8450	0.6525	0.7325	0.6975	0.8150	0.0000
r_{21}	0.3550	0.6400	0.4400	0.0000	0.5975	0.0000	0.0000	0.6500	0.0000	0.0000	0.6700	0.0000	0.0000	0.6650	0.6950
r_{22}	0.4850	0.6400	0.4500	0.6050	0.6975	0.0000	0.0000	0.0000	0.0000	0.0000	0.6475	0.7625	0.6625	0.3775	0.0000
r_{23}	0.7850	0.0000	0.0000	0.5725	0.0000	0.0000	0.6375	0.0000	0.0000	0.8350	0.7275	0.6750	0.0000	0.0000	0.0000
r_{24}	0.4050	0.0000	0.0000	0.0000	0.0000	0.3325	0.6275	0.0000	0.0000	0.0000	0.0000	0.5700	0.6775	0.6475	0.0000
r_{31}	0.0000	0.5800	0.6300	0.7150	0.4475	0.4600	0.0000	0.0000	0.5850	0.0000	0.0000	0.6925	0.6875	0.7950	0.0000
r_{32}	0.0000	0.0000	0.6000	0.0000	0.0000	0.5200	0.0000	0.0000	0.0000	0.0000	0.6525	0.6750	0.6875	0.7325	0.6475
r_{33}	0.0000	0.7000	0.6150	0.6500	0.5075	0.6350	0.0000	0.0000	0.6550	0.0000	0.7250	0.6175	0.7100	0.6075	0.6125

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 24. Pesos finales y pesos finales normalizados

	Procesos														
	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}
Pesos Finales	0.2000	0.1330	0.2000	0.1330	0.1330	0.2000	0.0670	0.2000	0.1330	0.0670	0.0670	0.2000	0.0670	0.0670	0.2000
Pesos Finales Normalizados	0.0970	0.0650	0.0970	0.0650	0.0650	0.0970	0.0320	0.0970	0.0650	0.0320	0.0320	0.0970	0.0320	0.0320	0.0970

Nota: Pesos finales y pesos finales normalizados asignados a los procesos para calcular la prioridad o preferencia final de acceso a los recursos. Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 25. Prioridades globales finales de los procesos para acceder a cada recurso

Recursos	Prioridades Nodales de los Procesos														
	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}
r_{11}	0.0690	0.0530	0.0690	0.0000	0.0000	0.0000	0.0200	0.0000	0.0000	0.0240	0.0220	0.0000	0.0000	0.0230	0.0700
r_{12}	0.0480	0.0400	0.0520	0.0270	0.0000	0.0580	0.0220	0.0000	0.0000	0.0210	0.0250	0.0650	0.0170	0.0240	0.0740
r_{13}	0.0000	0.0000	0.0720	0.0390	0.0000	0.0000	0.0000	0.0000	0.0500	0.0270	0.0210	0.0710	0.0230	0.0260	0.0000
r_{21}	0.0340	0.0410	0.0430	0.0000	0.0390	0.0000	0.0000	0.0630	0.0000	0.0000	0.0220	0.0000	0.0000	0.0210	0.0670
r_{22}	0.0470	0.0410	0.0440	0.0390	0.0450	0.0000	0.0000	0.0000	0.0000	0.0000	0.0210	0.0740	0.0210	0.0120	0.0000
r_{23}	0.0760	0.0000	0.0000	0.0370	0.0000	0.0000	0.0210	0.0000	0.0000	0.0270	0.0230	0.0650	0.0000	0.0000	0.0000
r_{24}	0.0390	0.0000	0.0000	0.0000	0.0000	0.0320	0.0200	0.0000	0.0000	0.0000	0.0000	0.0550	0.0220	0.0210	0.0000
r_{31}	0.0000	0.0370	0.0610	0.0460	0.0290	0.0450	0.0000	0.0000	0.0380	0.0000	0.0000	0.0670	0.0220	0.0260	0.0000
r_{32}	0.0000	0.0000	0.0580	0.0000	0.0000	0.0500	0.0000	0.0000	0.0000	0.0000	0.0210	0.0650	0.0220	0.0240	0.0630
r_{33}	0.0000	0.0450	0.0600	0.0420	0.0330	0.0610	0.0000	0.0000	0.0420	0.0000	0.0230	0.0600	0.0230	0.0200	0.0600

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

El siguiente paso es reiterar el procedimiento, pero retirando de las solicitudes de recursos las asignaciones ya hechas; también debe tenerse en cuenta que los recursos asignados quedarán disponibles cuando los procesos los hayan liberado, pudiendo por lo tanto ser asignados a otros procesos; el resultado de las sucesivas iteraciones se muestra en las Tabla 26, Tabla 27, Tabla 28, Tabla 29, Tabla 30, Tabla 31, Tabla 32, Tabla 33, Tabla 34, Tabla 35, Tabla 36.

TABLA 26. Orden o prioridad final (segunda iteración)

Prioridad Global Final Ordenada	Asignación
0.40653226	r_{33} al p_{34}
0.39951613	r_{12} al p_{34}
0.30346774	r_{31} al p_{13}
0.28153226	r_{11} al p_{11}
0.27024194	r_{22} al p_{11}
0.26274194	r_{21} al p_{25}
0.25701613	r_{13} al p_{34}
0.23790323	r_{32} al p_{37}
0.17322581	r_{23} al p_{34}
0.13435484	r_{24} al p_{11}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 27. Orden o prioridad final de asignación (tercera iteración)

Prioridad Global Final Ordenada	Asignación
0.34677419	r_{33} al p_{13}
0.33443548	r_{12} al p_{23}
0.24250000	r_{31} al p_{21}
0.22330645	r_{22} al p_{13}
0.21233871	r_{11} al p_{13}
0.19983871	r_{21} al p_{13}
0.18612903	r_{13} al p_{31}
0.17524194	r_{32} al p_{13}
0.10790323	r_{23} al p_{21}
0.09516129	r_{24} al p_{23}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 28. Orden o prioridad final de asignación (cuarta iteración)

Prioridad Global Final Ordenada	Asignación
0.28725806	r_{33} al p_{37}
0.27637097	r_{12} al p_{13}
0.19637097	r_{31} al p_{23}
0.17975806	r_{22} al p_{12}
0.15725806	r_{21} al p_{12}
0.14314516	r_{11} al p_{12}
0.13629032	r_{13} al p_{21}
0.11717742	r_{32} al p_{23}
0.07096774	r_{23} al p_{32}
0.06298387	r_{24} al p_{35}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 29. Orden o prioridad final de asignación (quinta iteración)

Prioridad Global Final Ordenada	Asignación
0.22798387	r_{33} al p_{12}
0.22459677	r_{12} al p_{11}
0.15185484	r_{31} al p_{31}
0.13846774	r_{22} al p_{21}

TABLA 29. Orden o prioridad final de asignación (quinta iteración)

Prioridad Global Final Ordenada	Asignación
0.11596774	r_{21} al p_{22}
0.09709677	r_{13} al p_{32}
0.08991935	r_{11} al p_{32}
0.06685484	r_{32} al p_{36}
0.04403226	r_{23} al p_{33}
0.04112903	r_{24} al p_{36}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 30. Orden o prioridad final de asignación (sexta iteración)

Prioridad Global Final Ordenada	Asignación
0.18282258	r_{33} al p_{31}
0.17669355	r_{12} al p_{12}
0.11411290	r_{31} al p_{12}
0.09943548	r_{22} al p_{22}
0.07741935	r_{21} al p_{11}
0.06983871	r_{13} al p_{36}
0.06604839	r_{11} al p_{36}
0.04322581	r_{32} al p_{35}
0.02056452	r_{23} al p_{24}
0.02024194	r_{24} al p_{24}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 31. Orden o prioridad final de asignación (séptima iteración)

Prioridad Global Final Ordenada	Asignación
0.14056452	r_{33} al p_{21}
0.13669355	r_{12} al p_{21}
0.07669355	r_{31} al p_{22}
0.05443548	r_{22} al p_{35}
0.04354839	r_{13} al p_{35}
0.04306452	r_{21} al p_{33}
0.04266129	r_{11} al p_{33}
0.02104839	r_{32} al p_{33}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 32. Orden o prioridad final de asignación (octava iteración)

Prioridad Global Final Ordenada	Asignación
0.10975806	r_{12} al p_{33}
0.09862903	r_{33} al p_{22}
0.04782258	r_{31} al p_{36}
0.03306452	r_{22} al p_{33}
0.02145161	r_{21} al p_{36}
0.02104839	r_{13} al p_{33}
0.02032258	r_{11} al p_{24}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 33. Orden o prioridad final de asignación (novena iteración)

Prioridad Global Final Ordenada	Asignación
0.08443548	r_{12} al p_{36}
0.06588710	r_{33} al p_{33}
0.02217742	r_{31} al p_{35}
0.01217742	r_{22} al p_{36}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 34. Orden o prioridad final de asignación (décima iteración)

Prioridad Global Final Ordenada	Asignación
0.06032258	r_{12} al p_{24}
0.04250000	r_{33} al p_{35}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 35. Orden o prioridad final de asignación (décimo primera iteración)

Prioridad Global Final Ordenada	Asignación
0.03798387	r_{12} al p_{32}
0.01959677	r_{33} al p_{36}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 36. Orden o prioridad final de asignación (décimo segunda iteración)

Prioridad Global Final Ordenada	Asignación
0.01693548	r_{12} al p_{35}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

De esta manera se han atendido 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.

2.6. Evaluación

El modelo propuesto logra que el sistema distribuido se auto regule reiteradamente en función del estado local de los n nodos, produciéndose una actualización de los estados locales de los mismos como consecuencia de la evolución de sus respectivos procesos y de las decisiones de acceso a los recursos: el sistema distribuido en el que se ejecutan grupos de procesos que acceden a recursos críticos, se observa a sí mismo y produce decisiones de accesos a recursos que modifican el estado del sistema y lo reajustan reiterativamente, garantizándose además la exclusión mutua en el acceso a los recursos compartidos, indicándose la prioridad de otorgamiento de acceso a cada recurso y el proceso al cual se lo asigna; este proceso se repite mientras haya procesos que soliciten acceso a recursos compartidos.

2.7. Discusiones y comentarios

El modelo propuesto es un modelo general, el agrupamiento de procesos no interfiere en la decisión final de asignaciones de recursos, tampoco contempla ningún tipo de consenso para lograr el acceso a los recursos.

El modelo propuesto incluye como caso particular uno de los métodos más utilizados, consistente en considerar sólo la prioridad de los procesos, en vez de un

grupo de variables de estado de cada nodo.

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

El contenido de este capítulo sirvió de base para las siguientes publicaciones:

- “Modelo de Asignación de Recursos para la Enseñanza de los Procesos Distribuidos”; D. L. la Red Martínez; F. Agostini; Primer Congreso Latinoamericano de Ingeniería - CLADI - 2017; Centro Provincial de Convenciones Paraná, Entre Ríos, Argentina.
- “Assignment of Resources in Distributed Systems”; D. L. La Red Martínez, J. C. Acosta, F. Agostini. 9th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2018), Orlando, USA.
- “Nueva propuesta para la administración de recursos y procesos en sistemas distribuidos”. D. L. La Red Martínez; F. Agostini; C. Primorac; Workshop de Investigadores en Ciencias de la Computación; WICC; 2018; ISBN 978-987-3619-27-4; Corrientes, Argentina.

Capítulo III - Operador de agregación definido para el escenario 2 (E2)

3.1. Introducción

Trabajos previos

En (La Red Martínez, 2017) y (La Red Martínez et al. 2018) se desarrollan operadores de agregación para la asignación de recursos en sistemas distribuidos.

3.2. Hipótesis y objetivos

Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua **puediendo constituir grupos** de procesos (los procesos independientes son considerados grupos unitarios); los procesos no requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y **con exigencias estrictas de consenso** para lograr el acceso (se requiere 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 puede ser interrumpida para asignar recursos a otro proceso).

3.3. Propuesta de solución

Se presentará una variante de un método innovador para la gestión de recursos compartidos en sistemas distribuidos, basado en (La Red Martínez, 2017) y (La Red Martínez et al. 2018), donde se desarrolla un operador de agregación, para asignar recursos en sistemas distribuidos. En este trabajo se establece un modelo de consenso que favorece el acceso secuencial de los procesos a todos los recursos solicitados. Las premisas, las estructuras de datos y el operador mencionado en el capítulo anterior se utilizan como punto de partida para crear un nuevo operador en el escenario descrito a continuación.

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 ser capaces de formar grupos de procesos (los

procesos independientes se consideran 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 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, hasta que el proceso activo libere los recursos).

Se trata de grupos de procesos distribuidos en nodos de procesos que acceden a recursos críticos compartidos en la modalidad de exclusión mutua distribuida, debiendo decidirse, ante la demanda de recursos por parte de los procesos, cuáles serán las prioridades para asignar los recursos a los procesos que los requieren (sólo intervendrán como alternativas de asignación a los procesos aquellos recursos disponibles, es decir, no asignados aún a determinados procesos):

- El permiso de acceso a los recursos compartidos propios de un nodo no dependerá sólo de si los nodos los están utilizando o no, sino del valor de agregación de las preferencias (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas).
- Las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas) dependerá de la consideración del valor de variables que representen el estado de cada uno de los distintos nodos. Cada nodo deberá expresar sus prioridades para la asignación de los distintos recursos compartidos respecto de los requerimientos de recursos de cada proceso.
- Consenso, que se entiende por priorizar al proceso que resulte con mayor prioridad global promedio en sus asignaciones, según las tablas de asignaciones.

Fig. 1 muestra los requerimientos de recursos por parte de los procesos, los recursos ya asignados y en que nodos se encuentran alojados.

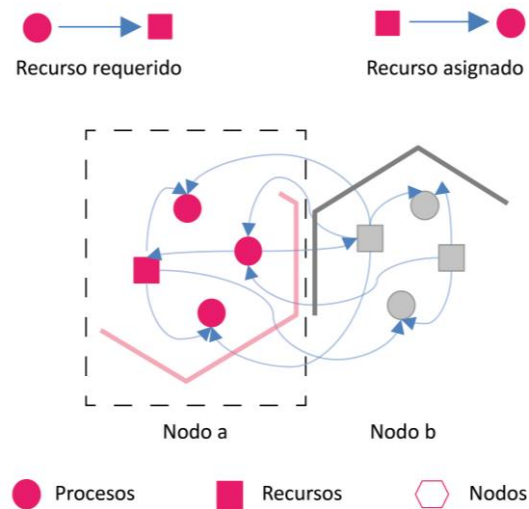


Fig. 1. Recursos y procesos en nodos en sistemas distribuidos
Fuente: Elaboración propia.

3.4. Descripción del operador de agregación

El operador propuesto consta de las siguientes etapas:

- 1) Cálculo de la carga computacional actual de los nodos.
- 2) Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas.
- 3) 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).
- 4) 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.
- 5) Cálculo de las prioridades o preferencias de los procesos para acceder secuencialmente, a todos sus recursos compartidos.

Cada uno de los pasos de la Fig. 2 fueron desarrollados en (La Red Martínez, 2017) y (La Red Martínez et al. 2018).

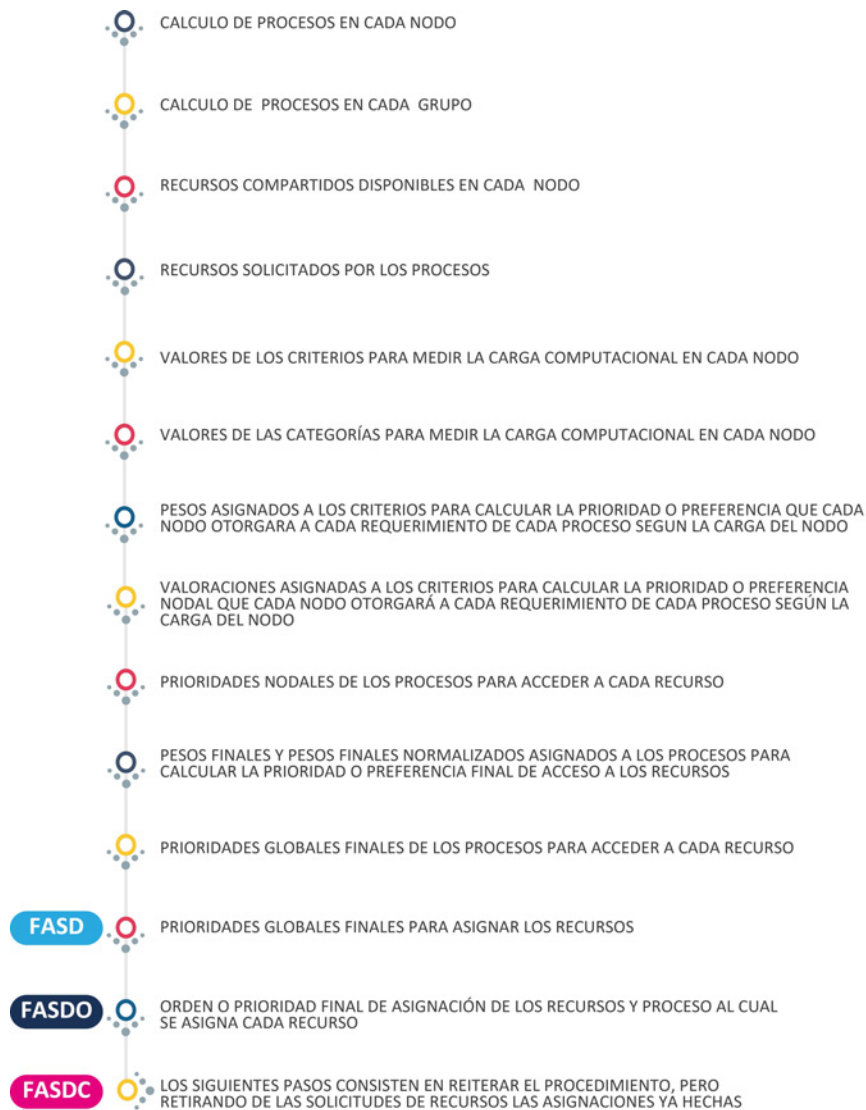


Fig. 2. Pasos para obtener las funciones **FASD**, **FASDO** y **FASDC**
 Fuente: Elaboración propia.

En la Fig. 2 se muestra una lista de los pasos necesarios para obtener las prioridades globales definitivas para asignar los recursos (**Función de Asignación para Sistemas Distribuidos - FASD**). El orden o prioridad de asignación de los recursos y el proceso al que se asigna cada recurso (**Función de Asignación para Sistemas Distribuidos Ordenada - FASDO**) puede verse en la Tabla 1. La unión de todas las asignaciones ordenadas, realizadas en todas las rondas (**FASDO**) conforman lo que se denomina (**Función de Asignación para Sistemas Distribuidos Concatenada - FASDC**), como se muestra en la Fig. 3.

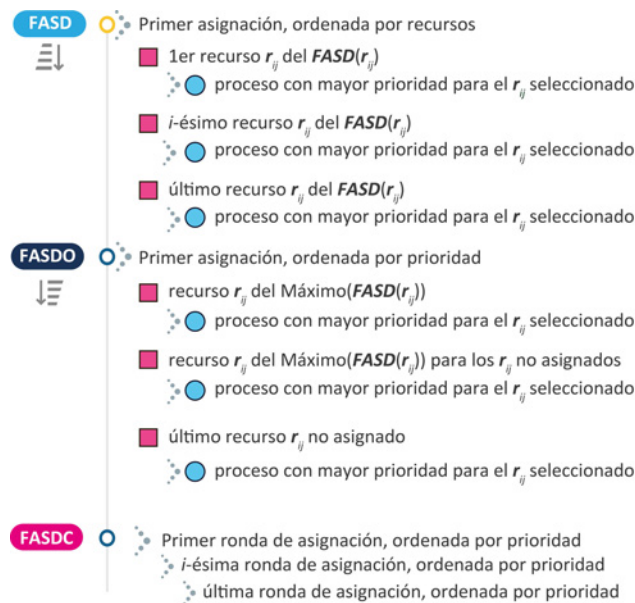


Fig. 3. Pasos para la **FASD**, **FASDO** y **FASDC** en cada iteración
Fuente: Elaboración propia.

La tabla **FASDC**, se obtendrá a partir de la concatenación de las tablas **FASDO** de cada iteración, según se muestra en la Tabla 37.

TABLA 37. Concatenación de las tablas de asignación ordenada (**FASDC**)

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

Nota: Concatenación de las tablas de asignación ordenada (FASDO) de cada una de las iteraciones del método general.
Fuente: Elaboración propia.

Prioridad global final del proceso

Una vez completada la tabla **FASDC** (Tabla 37), 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. Para ello, se tendrá en cuenta la tabla **FASDC**, 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 del Proceso (PGFP)**, como se muestra en Fig. 4:

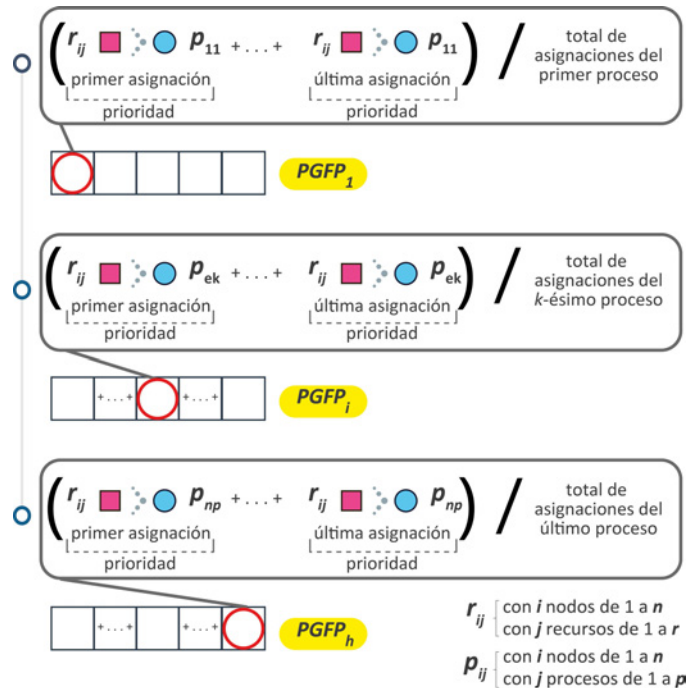


Fig. 4. Cálculo de la **PGFP** de cada proceso
Fuente: Elaboración propia.

$$PGFP_i = i = 1, \dots, h = \frac{\sum FASDC_j}{\text{Cantidad de asignaciones global del proceso } i} \quad h = \text{total de procesos en el sistema (sumatoria de procesos de los nodos); } j = n^\circ \text{ de recursos asignados al proceso } i.$$

Del vector **PGFP** 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 Fig. 5.

PGFPO_i = Prioridad Global Final del Proceso Ordenada

j = cardinalidad de **PGFP** (total de procesos en el sistema)

PGFPO_i = Max (**PGFP**_i no ordenado) $i = 1, \dots, j$

No ordenado = **PGFP**_i \notin **PGFPO**

1ero: **PGFPO**_i = Max (**PGFP**_i) $i = 1, \dots, j$

2do: $PGFPO_2 = \text{Max} (\text{no ordenado } PGFP_i) \quad i= 1, \dots, j$

...

último: $PGFPO_j = \text{Max} (\text{no ordenado } PGFP_i) \quad i= 1, \dots, j$

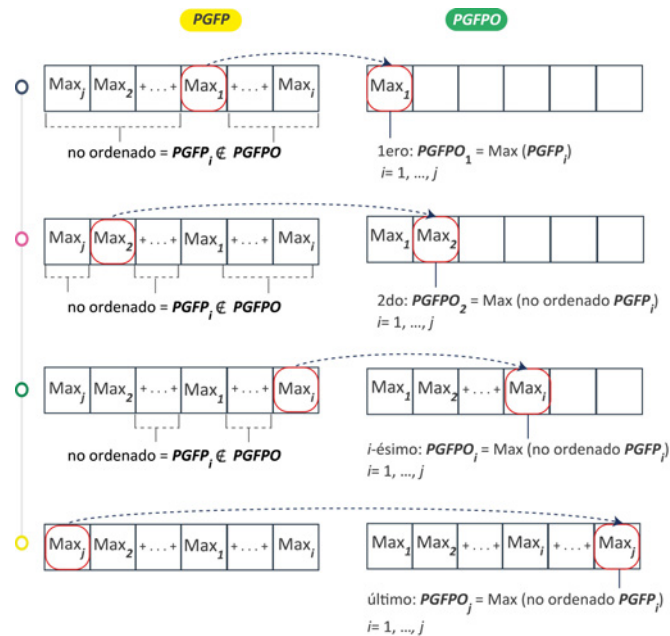


Fig. 5. Ejemplo para calcular la PGFPO de cada proceso
Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Concatenada Ordenada (FASDCO)

La **FASDCO** 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 **FASDC** y la **PGFPO**, como se muestra en la Fig. 6.

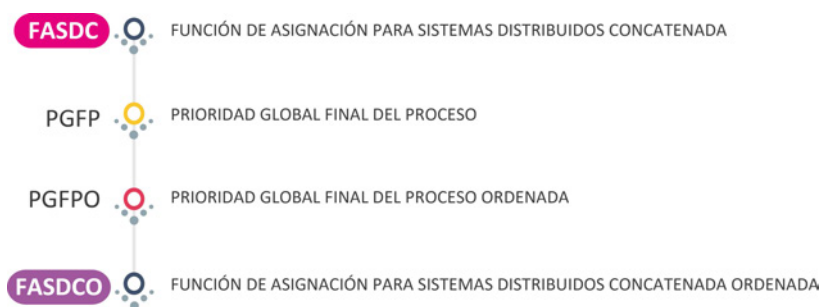


Fig. 6. Pasos para obtener desde la FASDC a la FASDCO
Fuente: Elaboración propia.

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

$cp_i =$ cardinalidad proceso(**PGFPO**) _{i} en **FASDC**

Luego, habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla **FASDC** de cada uno de los procesos del vector **PGFPO**. El total de elementos por cada proceso, estará determinado por la cardinalidad calculada en el paso anterior, como se puede ver en Fig. 7.

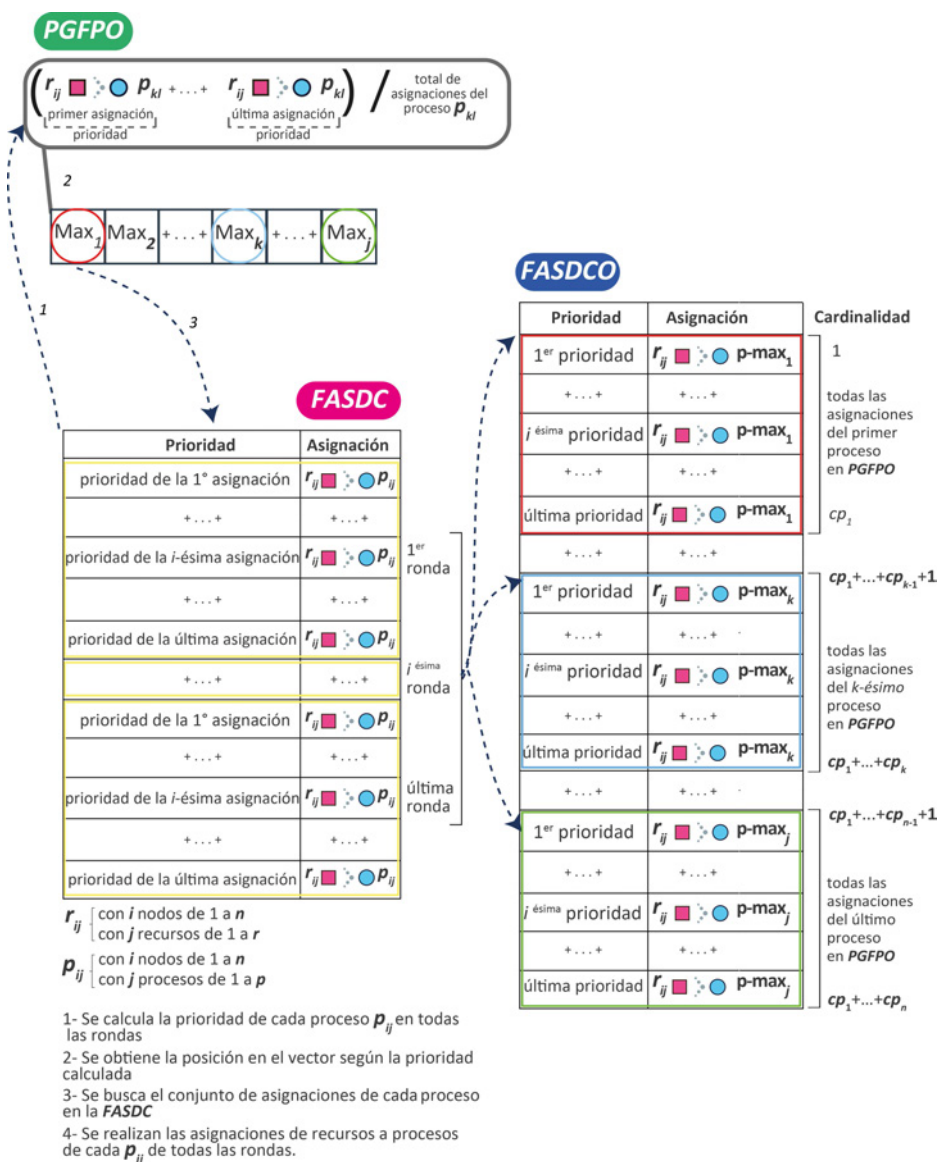


Fig. 7. Esquema para obtener la **FASDC** de cada proceso
Fuente: Elaboración propia.

$FASDCO_1$ = primer elemento de la **FASDC** para el proceso(**PGFPO**₁)

...

$FASDCO_{cp1}$ = último elemento de la **FASDC** para el proceso(**PGFPO**₁)

$FASDCO_{cp1+1}$ = primer elemento de la **FASDC** para el proceso(**PGFPO**₂)

...

$FASDCO_{cp1+cp2}$ = último elemento de la **FASDC** para el proceso(**PGFPO**₂)

...

...

$FASDCO_{cp1+cp2+\dots+cpk-1+1}$ = primer elemento de la **FASDC** para el proceso(**PGFPO**_k)

...

$FASDCO_{cp1+cp2+\dots+cpk}$ = último elemento de la **FASDC** para el proceso(**PGFPO**_k)

$FASDCO_{cp1+cp2+\dots+cpn-1+1}$ = primer elemento de la **FASDC** para el proceso(**PGFPO**_n)

...

$FASDCO_{cp1+cp2+\dots+cpn}$ = último elemento de la **FASDC** para el proceso(**PGFPO**_n)

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.

3.5. Ejemplo

En esta sección se explicará en detalle un ejemplo de aplicación del operador de agregación propuesto. Este ejemplo toma como punto de partida el vector **FASD** ordenado de (La Red Martínez, 2017) y (La Red Martínez et al. 2018) y estos pasos se muestran en la Fig. 2. Todas las figuras y tablas que consignan fuente de elaboración propia, basado en La Red Martínez (2017), se transcriben en el ejemplo del escenario (E1).

El ejemplo visto en el (La Red Martínez, 2017) muestra los siguientes cálculos:

- Las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles.
- El vector de pesos finales que se utilizará en el proceso de agregación final para determinar el orden o prioridad de acceso a los recursos.

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 tendrá dicho recurso para ser asignado. Esto constituye la Función de Asignación de Sistemas Distribuidos (**FASD**), que se puede ver en la Tabla 38.

TABLA 38. *Prioridad final global (FASD) primera iteración*

Recursos	Prioridad	Asignación
r_{11}	0.35120968	r_{11} al p_{37}
r_{12}	0.47306452	r_{12} al p_{37}
r_{13}	0.32862903	r_{13} al p_{13}
r_{21}	0.33000000	r_{21} al p_{37}
r_{22}	0.34403226	r_{22} al p_{34}
r_{23}	0.24919355	r_{23} al p_{11}
r_{24}	0.18951613	r_{24} al p_{34}
r_{31}	0.37048387	r_{31} al p_{34}
r_{32}	0.30322581	r_{32} al p_{34}
r_{33}	0.46798387	r_{33} al p_{23}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

El orden final de asignación de los recursos y de los procesos receptores se obtiene ordenando la Tabla 38, como se muestra en la Tabla 39.

TABLA 39. *Orden final de asignación (FASDO) primer iteración*

Prioridad Global Final Ordenada	Asignación
0.47306452	r_{12} al p_{37}
0.46798387	r_{33} al p_{23}
0.37048387	r_{31} al p_{34}
0.35120968	r_{11} al p_{37}
0.34403226	r_{22} al p_{34}
0.33000000	r_{21} al p_{37}
0.32862903	r_{13} al p_{13}
0.30322581	r_{32} al p_{34}
0.24919355	r_{23} al p_{11}
0.18951613	r_{24} al p_{34}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de

asignaciones ya realizadas; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos.

De esta manera, se dará respuesta 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, según el escenario presentado en (La Red Martínez, 2017) y (La Red Martínez et al. 2018).

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 **FASDC** se obtendrá a partir de la concatenación de la tabla **FASDO** de cada iteración, como se muestra en la Tabla 40.

TABLA 40. Orden final de asignación todas las iteraciones (**FASDC**)

Prioridad Final Ordenada	Asignación	Ronda
0.47306452	r_{12} al p_{37}	1
0.46798387	r_{33} al p_{23}	1
0.37048387	r_{31} al p_{34}	1
0.35120968	r_{11} al p_{37}	1
0.34403226	r_{22} al p_{34}	1
0.33000000	r_{21} al p_{37}	1
0.32862903	r_{13} al p_{13}	1
0.30322581	r_{32} al p_{34}	1
0.24919355	r_{23} al p_{11}	1
0.18951613	r_{24} al p_{34}	1
0.40653226	r_{33} al p_{34}	2
0.39951613	r_{12} al p_{34}	2
0.30346774	r_{31} al p_{13}	2
0.28153226	r_{11} al p_{11}	2
0.27024194	r_{22} al p_{11}	2
0.26274194	r_{21} al p_{25}	2
0.25701613	r_{13} al p_{34}	2
0.23790323	r_{32} al p_{37}	2
0.17322581	r_{23} al p_{34}	2
0.13435484	r_{24} al p_{11}	2
0.34677419	r_{33} al p_{13}	3
0.33443548	r_{12} al p_{23}	3
0.24250000	r_{31} al p_{21}	3
0.22330645	r_{22} al p_{13}	3
0.21233871	r_{11} al p_{13}	3
0.19983871	r_{21} al p_{13}	3
0.18612903	r_{13} al p_{31}	3
0.17524194	r_{32} al p_{13}	3
0.10790323	r_{23} al p_{21}	3
0.09516129	r_{24} al p_{23}	3
0.28725806	r_{33} al p_{37}	4
0.27637097	r_{12} al p_{13}	4
0.19637097	r_{31} al p_{23}	4

TABLA 40. Orden final de asignación todas las iteraciones (**FASDC**)

Prioridad Final Ordenada	Asignación	Ronda
0.17975806	r_{22} al p_{12}	4
0.15725806	r_{21} al p_{12}	4
0.14314516	r_{11} al p_{12}	4
0.13629032	r_{13} al p_{21}	4
0.11717742	r_{32} al p_{23}	4
0.07096774	r_{23} al p_{32}	4
0.06298387	r_{24} al p_{35}	4
0.22798387	r_{33} al p_{12}	5
0.22459677	r_{12} al p_{11}	5
0.15185484	r_{31} al p_{31}	5
0.13846774	r_{22} al p_{21}	5
0.11596774	r_{21} al p_{22}	5
0.09709677	r_{13} al p_{32}	5
0.08991935	r_{11} al p_{32}	5
0.06685484	r_{32} al p_{36}	5
0.04403226	r_{23} al p_{33}	5
0.04112903	r_{24} al p_{36}	5
0.18282258	r_{33} al p_{31}	6
0.17669355	r_{12} al p_{12}	6
0.11411290	r_{31} al p_{12}	6
0.09943548	r_{22} al p_{22}	6
0.07741935	r_{21} al p_{11}	6
0.06983871	r_{13} al p_{36}	6
0.06604839	r_{11} al p_{36}	6
0.04322581	r_{32} al p_{35}	6
0.02056452	r_{23} al p_{24}	6
0.02024194	r_{24} al p_{24}	6
0.14056452	r_{33} al p_{21}	7
0.13669355	r_{12} al p_{21}	7
0.07669355	r_{31} al p_{22}	7
0.05443548	r_{22} al p_{35}	7
0.04354839	r_{13} al p_{35}	7
0.04306452	r_{21} al p_{33}	7
0.04266129	r_{11} al p_{33}	7
0.02104839	r_{32} al p_{33}	7
0.10975806	r_{12} al p_{33}	8
0.09862903	r_{33} al p_{22}	8
0.04782258	r_{31} al p_{36}	8
0.03306452	r_{22} al p_{33}	8
0.02145161	r_{21} al p_{36}	8
0.02104839	r_{13} al p_{33}	8
0.02032258	r_{11} al p_{24}	8
0.08443548	r_{12} al p_{36}	9
0.06588710	r_{33} al p_{33}	9
0.02217742	r_{31} al p_{35}	9
0.01217742	r_{22} al p_{36}	9
0.06032258	r_{12} al p_{24}	10
0.04250000	r_{33} al p_{35}	10
0.03798387	r_{12} al p_{32}	10
0.01959677	r_{33} al p_{36}	10
0.01693548	r_{12} al p_{35}	11

Nota: Orden o prioridad final de asignación de recursos a procesos al que se le asigna cada recurso en todas las iteraciones (**FASDC**).

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDC**, se calcularán las Prioridades Finales Globales del Proceso (**PGFP**):

$$PGFP_1 = (0.24919355 + 0.28153226 + 0.27024194 + 0.13435484 + 0.22459677 + 0.07741935) / 6$$

$$PGFP_2 = (0.17975806 + 0.15725806 + 0.14314516 + 0.22798387 + 0.17669355 + 0.11411290) / 6$$

$$PGFP_3 = (0.32862903 + 0.30346774 + 0.34677419 + 0.22330645 + 0.21233871 + 0.19983871 + 0.17524194 + 0.27637097) / 7$$

$$PGFP_4 = (0.24250000 + 0.10790323 + 0.13629032 + 0.13846774 + 0.14056452 + 0.13669355) / 6$$

$$PGFP_5 = (0.11596774 + 0.09943548 + 0.07669355 + 0.09862903) / 4$$

$$PGFP_6 = (0.46798387 + 0.33443548 + 0.09516129 + 0.19637097 + 0.11717742) / 5$$

$$PGFP_7 = (0.02056452 + 0.02024194 + 0.02032258 + 0.06032258) / 4$$

$$PGFP_8 = 0.26274194 / 1$$

$$PGFP_9 = (0.18612903 + 0.15185484 + 0.18282258) / 3$$

$$PGFP_{10} = (0.07096774 + 0.09709677 + 0.08991935 + 0.03798387) / 4$$

$$PGFP_{11} = (0.04403226 + 0.04306452 + 0.04266129 + 0.02104839 + 0.10975806 + 0.03306452 + 0.02104839 + 0.06588710) / 8$$

$$PGFP_{12} = (0.37048387 + 0.34403226 + 0.30322581 + 0.18951613 + 0.40653226 + 0.39951613 + 0.25701613 + 0.17322581) / 8$$

$$PGFP_{13} = (0.06298387 + 0.04322581 + 0.05443548 + 0.04354839 + 0.02217742 + 0.04250000 + 0.01693548) / 7$$

$$PGFP_{14} = (0.06685484 + 0.04112903 + 0.06983871 + 0.06604839 + 0.04782258 + 0.02145161 + 0.08443548 + 0.01217742 + 0.01959677) / 9$$

$$PGFP_{15} = (0.47306452 + 0.35120968 + 0.33000000 + 0.23790323 + 0.28725806) / 5$$

La tabla de la **FASDC** ordenada por proceso, como se muestra en la Tabla 41.

TABLA 41. Prioridad Global Final Ordenada por proceso

Prioridad Final Global	Recursos	Procesos	Rondas
0.24919355	r23	p11	1
0.28153226	r11	p11	2
0.27024194	r22	p11	2
0.13435484	r24	p11	2
0.22459677	r12	p11	5
0.07741935	r21	p11	6
0.17975806	r22	p12	4
0.15725806	r21	p12	4
0.14314516	r11	p12	4
0.22798387	r33	p12	5
0.17669355	r12	p12	6
0.11411290	r31	p12	6
0.32862903	r13	p13	1
0.30346774	r31	p13	2
0.34677419	r33	p13	3
0.22330645	r22	p13	3
0.21233871	r11	p13	3

TABLA 41. Prioridad Global Final Ordenada por proceso

Prioridad Final Global	Recursos	Procesos	Rondas
0.19983871	r_{21}	p_{13}	3
0.17524194	r_{32}	p_{13}	3
0.27637097	r_{12}	p_{13}	4
0.24250000	r_{31}	p_{21}	3
0.10790323	r_{23}	p_{21}	3
0.13629032	r_{13}	p_{21}	4
0.13846774	r_{22}	p_{21}	5
0.14056452	r_{33}	p_{21}	7
0.13669355	r_{12}	p_{21}	7
0.11596774	r_{21}	p_{22}	5
0.09943548	r_{22}	p_{22}	6
0.07669355	r_{31}	p_{22}	7
0.09862903	r_{33}	p_{22}	8
0.46798387	r_{33}	p_{23}	1
0.33443548	r_{12}	p_{23}	3
0.09516129	r_{24}	p_{23}	3
0.19637097	r_{31}	p_{23}	4
0.11717742	r_{32}	p_{23}	4
0.02056452	r_{23}	p_{24}	6
0.02024194	r_{24}	p_{24}	6
0.02032258	r_{11}	p_{24}	8
0.06032258	r_{12}	p_{24}	10
0.26274194	r_{21}	p_{25}	2
0.18612903	r_{13}	p_{31}	3
0.15185484	r_{31}	p_{31}	5
0.18282258	r_{33}	p_{31}	6
0.07096774	r_{23}	p_{32}	4
0.09709677	r_{13}	p_{32}	5
0.08991935	r_{11}	p_{32}	5
0.03798387	r_{12}	p_{32}	10
0.04403226	r_{23}	p_{33}	5
0.04306452	r_{21}	p_{33}	7
0.04266129	r_{11}	p_{33}	7
0.02104839	r_{32}	p_{33}	7
0.10975806	r_{12}	p_{33}	8
0.03306452	r_{22}	p_{33}	8
0.02104839	r_{13}	p_{33}	8
0.06588710	r_{33}	p_{33}	9
0.37048387	r_{31}	p_{34}	1
0.34403226	r_{22}	p_{34}	1
0.30322581	r_{32}	p_{34}	1
0.18951613	r_{24}	p_{34}	1
0.40653226	r_{33}	p_{34}	2
0.39951613	r_{12}	p_{34}	2
0.25701613	r_{13}	p_{34}	2
0.17322581	r_{23}	p_{34}	2
0.06298387	r_{24}	p_{35}	4
0.04322581	r_{32}	p_{35}	6
0.05443548	r_{22}	p_{35}	7
0.04354839	r_{13}	p_{35}	7
0.02217742	r_{31}	p_{35}	9
0.04250000	r_{33}	p_{35}	10
0.01693548	r_{12}	p_{35}	11
0.06685484	r_{32}	p_{36}	5
0.04112903	r_{24}	p_{36}	5
0.06983871	r_{13}	p_{36}	6
0.06604839	r_{11}	p_{36}	6
0.04782258	r_{31}	p_{36}	8
0.02145161	r_{21}	p_{36}	8
0.08443548	r_{12}	p_{36}	9
0.01217742	r_{22}	p_{36}	9
0.01959677	r_{33}	p_{36}	10
0.47306452	r_{12}	p_{37}	1
0.35120968	r_{11}	p_{37}	1
0.33000000	r_{21}	p_{37}	1

TABLA 41. Prioridad Global Final Ordenada por proceso

Prioridad Final Global	Recursos	Procesos	Rondas
0.23790323	r_{32}	p_{37}	2
0.28725806	r_{33}	p_{37}	4

Fuente: Elaboración propia.

Al calcular el **PGFP** para todos los procesos, como se muestra en la Tabla 41, se obtendrá un vector, como se muestra en la Tabla 42.

TABLA 42. PGFP (prioridad global final del proceso)

Prioridad Global Final	Procesos
0.20622312	p_{11}
0.16649193	p_{12}
0.25824597	p_{13}
0.15040323	p_{21}
0.09768145	p_{22}
0.24222581	p_{23}
0.03036291	p_{24}
0.26274194	p_{25}
0.17360215	p_{31}
0.07399193	p_{32}
0.04757057	p_{33}
0.30544355	p_{34}
0.04082949	p_{35}
0.04770609	p_{36}
0.33588710	p_{37}

Fuente: Elaboración propia.

TABLA 43. PGFPO (prioridad global final del proceso ordenada)

Prioridad Global Final Ordenada	Procesos
0.33588710	p_{37}
0.30544355	p_{34}
0.26274194	p_{25}
0.25824597	p_{13}
0.24222581	p_{23}
0.20622312	p_{11}
0.17360215	p_{31}
0.16649193	p_{12}
0.15040323	p_{21}
0.09768145	p_{22}
0.07399193	p_{32}
0.04770609	p_{36}
0.04757057	p_{33}
0.04082949	p_{35}
0.03036291	p_{24}

Fuente: Elaboración propia.

Del vector **PGFP** 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 43.

Función de Asignación para Sistemas Distribuidos Concatenada Ordenada (**FASDCO**)

La **FASDCO** 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 **FASDC** y la **PGFPO**.

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

cp_i = cardinalidad proceso(**PGFPO**_{*i*}) en **FASDC**

cp_{37} = cardinalidad proceso(**PGFPO**₁) en **FASDC** = 5

cp_{34} = cardinalidad proceso(**PGFPO**₂) en **FASDC** = 8

cp_{25} = cardinalidad proceso(**PGFPO**₃) en **FASDC** = 1

cp_{13} = cardinalidad proceso(**PGFPO**₄) en **FASDC** = 8

cp_{23} = cardinalidad proceso(**PGFPO**₅) en **FASDC** = 5

cp_{11} = cardinalidad proceso(**PGFPO**₆) en **FASDC** = 6

cp_{31} = cardinalidad proceso(**PGFPO**₇) en **FASDC** = 3

cp_{12} = cardinalidad proceso(**PGFPO**₈) en **FASDC** = 6

cp_{21} = cardinalidad proceso(**PGFPO**₉) en **FASDC** = 6

cp_{22} = cardinalidad proceso(**PGFPO**₁₀) en **FASDC** = 4

cp_{32} = cardinalidad proceso(**PGFPO**₁₁) en **FASDC** = 4

cp_{36} = cardinalidad proceso(**PGFPO**₁₂) en **FASDC** = 9

cp_{33} = cardinalidad proceso(**PGFPO**₁₃) en **FASDC** = 8

cp_{35} = cardinalidad proceso(**PGFPO**₁₄) en **FASDC** = 7

cp_{24} = cardinalidad proceso(**PGFPO**₁₅) en **FASDC** = 4

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

FASDCO₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₁)

...

FASDCO₅ = último elemento de la **FASDC** para el proceso (**PGFPO**₁)

FASDCO₅₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₂)

...

FASDCO₅₊₈ = último elemento de la **FASDC** para el proceso (**PGFPO**₂)

FASDCO₅₊₈₊₁ = element de la **FASDC** para el proceso (**PGFPO**₃)

FASDCO₅₊₈₊₁₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₄)

...

FASDCO₅₊₈₊₁₊₈ = último elemento de la **FASDC** para el proceso (**PGFPO**₄)

FASDCO₅₊₈₊₁₊₈₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₅)

...

FASDCO₅₊₈₊₁₊₈₊₅ = último elemento de la **FASDC** para el proceso (**PGFPO**₅)

FASDCO₅₊₈₊₁₊₈₊₅₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₆)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆ = último elemento de la **FASDC** para el proceso (**PGFPO**₆)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₇)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃ = último elemento de la **FASDC** para el proceso (**PGFPO**₇)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₈)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆ = último elemento de la **FASDC** para el proceso (**PGFPO**₈)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₉)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆ = último elemento de la **FASDC** para el proceso (**PGFPO**₉)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₁₀)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄ = último elemento de la **FASDC** para el proceso (**PGFPO**₁₀)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₁₁)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄ = último elemento de la **FASDC** para el proceso (**PGFPO**₁₁)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₁₂)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄₊₉ = último elemento de la **FASDC** para el proceso (**PGFPO**₁₂)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄₊₉₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₁₃)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄₊₉₊₈ = último elemento de la **FASDC** para el proceso (**PGFPO**₁₃)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄₊₉₊₈₊₁ = primer elemento de la **FASDC** para el proceso (**PGFPO**₁₄)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄₊₉₊₈₊₇ = último elemento de la **FASDC** para el proceso (**PGFPO**₁₄)

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄₊₉₊₈₊₇₊₁ = primer elemento de la **FASDC** para el proceso

(PGFPO₁₅)

...

FASDCO₅₊₈₊₁₊₈₊₅₊₆₊₃₊₆₊₆₊₄₊₄₊₉₊₈₊₇₊₄ = último elemento de la **FASDC** para el proceso (PGFPO₁₅)

La Tabla 44 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 (**FASDCO**).

TABLA 44. Orden Final de asignación en la FASDCO

Prioridad	Recursos	Procesos	Rondas
0.4730645	r ₁₂	p ₃₇	1
0.3512097	r ₁₁	p ₃₇	1
0.3300000	r ₂₁	p ₃₇	1
0.2379032	r ₃₂	p ₃₇	2
0.2872581	r ₃₃	p ₃₇	4
0.3704839	r ₃₁	p ₃₄	1
0.3440323	r ₂₂	p ₃₄	1
0.3032258	r ₃₂	p ₃₄	1
0.1895161	r ₂₄	p ₃₄	1
0.4065323	r ₃₃	p ₃₄	2
0.3995161	r ₁₂	p ₃₄	2
0.2570161	r ₁₃	p ₃₄	2
0.1732258	r ₂₃	p ₃₄	2
0.2627419	r ₂₁	p ₂₅	2
0.3286290	r ₁₃	p ₁₃	1
0.3034677	r ₃₁	p ₁₃	2
0.3467742	r ₃₃	p ₁₃	3
0.2233065	r ₂₂	p ₁₃	3
0.2123387	r ₁₁	p ₁₃	3
0.1998387	r ₂₁	p ₁₃	3
0.1752419	r ₃₂	p ₁₃	3
0.2763710	r ₁₂	p ₁₃	4
0.4679839	r ₃₃	p ₂₃	1
0.3344355	r ₁₂	p ₂₃	3
0.0951613	r ₂₄	p ₂₃	3
0.1963710	r ₃₁	p ₂₃	4
0.1171774	r ₃₂	p ₂₃	4
0.2491936	r ₂₃	p ₁₁	1
0.2815323	r ₁₁	p ₁₁	2
0.2702419	r ₂₂	p ₁₁	2
0.1343548	r ₂₄	p ₁₁	2
0.2245968	r ₁₂	p ₁₁	5
0.0774194	r ₂₁	p ₁₁	6
0.1861290	r ₁₃	p ₃₁	3
0.1518548	r ₃₁	p ₃₁	5
0.1828226	r ₃₃	p ₃₁	6
0.1797581	r ₂₂	p ₁₂	4
0.1572581	r ₂₁	p ₁₂	4
0.1431452	r ₁₁	p ₁₂	4
0.2279839	r ₃₃	p ₁₂	5
0.1766936	r ₁₂	p ₁₂	6
0.1141129	r ₃₁	p ₁₂	6
0.2425000	r ₃₁	p ₂₁	3
0.1079032	r ₂₃	p ₂₁	3
0.1362903	r ₁₃	p ₂₁	4

TABLA 44. Orden Final de asignación en la FASDCO

Prioridad	Recursos	Procesos	Rondas
0.1384677	r_{22}	p_{21}	5
0.1405645	r_{33}	p_{21}	7
0.1366936	r_{12}	p_{21}	7
0.1159677	r_{21}	p_{22}	5
0.0994355	r_{22}	p_{22}	6
0.0766936	r_{31}	p_{22}	7
0.0986290	r_{33}	p_{22}	8
0.0709677	r_{23}	p_{32}	4
0.0970968	r_{13}	p_{32}	5
0.0899194	r_{11}	p_{32}	5
0.0379839	r_{12}	p_{32}	10
0.0668548	r_{32}	p_{36}	5
0.0411290	r_{24}	p_{36}	5
0.0698387	r_{13}	p_{36}	6
0.0660484	r_{11}	p_{36}	6
0.0478226	r_{31}	p_{36}	8
0.0214516	r_{21}	p_{36}	8
0.0844355	r_{12}	p_{36}	9
0.0121774	r_{22}	p_{36}	9
0.0195968	r_{33}	p_{36}	10
0.0440323	r_{23}	p_{33}	5
0.0430645	r_{21}	p_{33}	7
0.0426613	r_{11}	p_{33}	7
0.0210484	r_{32}	p_{33}	7
0.1097581	r_{12}	p_{33}	8
0.0330645	r_{22}	p_{33}	8
0.0210484	r_{13}	p_{33}	8
0.0658871	r_{33}	p_{33}	9
0.0629839	r_{24}	p_{35}	4
0.0432258	r_{32}	p_{35}	6
0.0544355	r_{22}	p_{35}	7
0.0435484	r_{13}	p_{35}	7
0.0221774	r_{31}	p_{35}	9
0.0425000	r_{33}	p_{35}	10
0.0169355	r_{12}	p_{35}	11
0.0205645	r_{23}	p_{24}	6
0.0202419	r_{24}	p_{24}	6
0.0203226	r_{11}	p_{24}	8
0.0603226	r_{12}	p_{24}	10

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.

3.6. 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 **PGFPO**, pero el orden en el que se deben realizar esas asignaciones, respeta el de la tabla **FASDC**, para cada proceso.

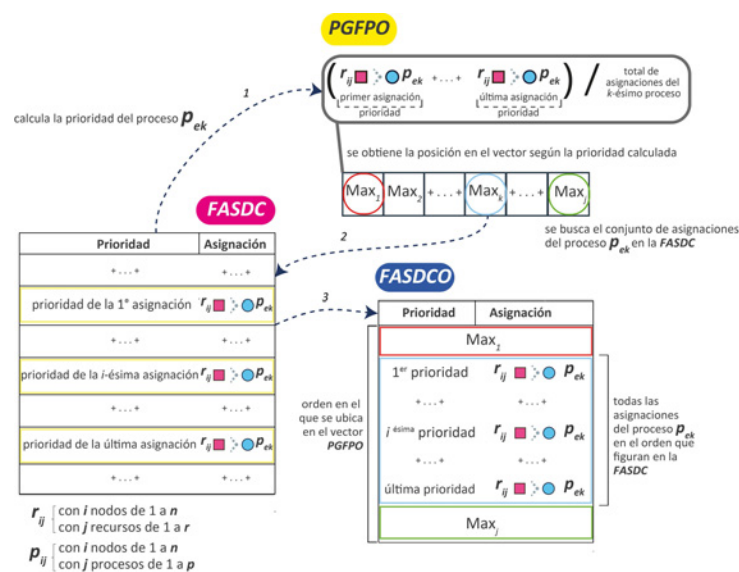


Fig. 8. Cálculo de prioridades para el proceso p_{ek} con mayor prioridad en **PGFPO**

Fuente: Elaboración propia.

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 Fig. 8.

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 **FASDCO** dependerá de la ubicación en el vector **PGFPO**.

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.

3.7. 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 (**FASDC**).

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.

En el siguiente capítulo, se desarrolla una variante del método propuesto

considerando otro operador de agregación para grupos de procesos que requieren requisitos de consenso estrictos.

El contenido de este capítulo sirvió de base para las siguientes publicaciones:

- 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)*. Vol. 9, N° 12. The Science and Information (SAI) Organization, England, U.K. <http://dx.doi.org/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”. 10th International Multi-Conference on Complexity, Informatics and Cybernetics (*IMCIC 2019*), Vol 1, Orlando, Florida, USA.

Capítulo IV - Operador de agregación definido para el escenario 3 (E3)

4.1. Introducción

Trabajos previos

En (La Red Martínez, 2017) y (La Red Martínez et al. 2018) se desarrollan operadores de agregación para la asignación de recursos en sistemas distribuidos.

4.2. Hipótesis y objetivos

Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua **constituyendo grupos** de procesos (los procesos independientes son considerados grupos unitarios); los procesos no requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y **con exigencias estrictas de consenso para lograr el acceso** (se requiere un consenso para asignar de manera consecutiva los recursos solicitados por un grupo de procesos, es decir, que iniciada la secuencia de asignación de recursos al grupo, la misma no puede ser interrumpida para asignar recursos a otro grupo).

4.3. Propuestas de solución

Se presentará una variante de un método innovador para la gestión de recursos compartidos en sistemas distribuidos, basado en (La Red Martínez, 2017) y (La Red Martínez et al. 2018), donde se desarrolla un operador de agregación, para asignar recursos en sistemas distribuidos, pero en este caso, estableciendo un modelo de consenso que favorezca el acceso secuencial de los procesos de cada grupo a todos los recursos solicitados. Las premisas, las estructuras de datos y el operador mencionado en el segundo capítulo se utilizan como punto de partida para crear un nuevo operador en el escenario descrito a continuación.

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 ser capaces de formar grupos de procesos (los

procesos independientes se consideran grupos unitarios y con prioridad grupal cero), 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 acceder a los recursos (se requiere un acuerdo para asignar consecutivamente los recursos solicitados por todos los procesos de cada grupo, es decir, una vez iniciada la secuencia de asignación de recursos al grupo, no se puede interrumpir para otorgársela a otro grupos de procesos, hasta que el proceso activo en cada grupo libere los recursos).

Se trata de grupos de procesos distribuidos en nodos de procesos que acceden a recursos críticos compartidos en la modalidad de exclusión mutua distribuida, debiendo decidirse, ante la demanda de recursos por parte de los procesos, cuáles serán las prioridades para asignar los recursos a los grupos de procesos que los requieren (sólo intervendrán como alternativas de asignación a los procesos aquellos recursos disponibles, es decir, no asignados aún a determinados procesos):

- El permiso de acceso a los recursos compartidos propios de un nodo no dependerá sólo de si los nodos los están utilizando o no, sino del valor de agregación de las preferencias (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas).
- Las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas) dependerá de la consideración del valor de variables que representen el estado de cada uno de los distintos nodos. Cada nodo deberá expresar sus prioridades para la asignación de los distintos recursos compartidos respecto de los requerimientos de recursos de cada proceso.
- Consenso, que se entiende por priorizar al grupo de procesos que resulte con mayor prioridad global promedio en sus asignaciones, según las tablas de asignaciones.

4.4. Descripción del operador de agregación

El operador propuesto consta de las siguientes etapas:

1. Cálculo de la carga computacional actual de los nodos.
2. Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas.
3. 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).
4. 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.
5. Cálculo de las prioridades o preferencias de los grupos de 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 grupo de procesos será asignado cada recurso.

Partiendo de las prioridades globales definitivas para asignar los recursos (***Función de Asignación para Sistemas Distribuidos - FASD***) y estableciendo el orden o prioridad de asignación de los recursos y el proceso al que se asigna cada recurso (***Función de Asignación para Sistemas Distribuidos Ordenada - FASDO***), el siguiente paso, como se muestra en la Fig. 9, consiste en la unión de todas las asignaciones en todas las rondas (***Función de Asignación para Sistemas Distribuidos Concatenada - FASDC***), por último, se establecerá la ***Función de Asignación para Sistemas Distribuidos Concatenada Ordenada (FASDCO)***.

Una vez calculada la ***FASDCO***, se establecerá la **prioridad grupal** y el orden en el que se asignarán los recursos a procesos en los grupos. Para ellos se tendrá en

cuenta todos los procesos y recursos vinculados en todas las instancias. La cardinalidad de grupo se medirá por la cantidad de procesos que hay en cada grupo, en todas las instancias.

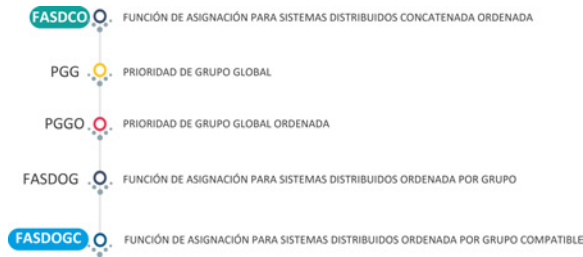
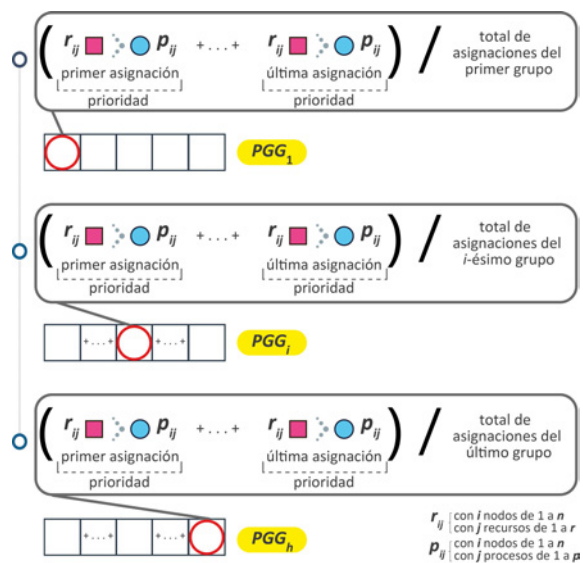


Fig. 9. Pasos para obtener desde la **FASDCO** a la **FASDOGC**
Fuente: Elaboración propia.

Prioridad del grupo en la asignación

La sumatoria de todas las prioridades finales (**FASDCO**) de cada relación proceso y recurso por grupo, dividido la cardinalidad (números de asignaciones de cada grupo, en todas las instancias) del grupo, indicará la prioridad que tendrá dicho grupo para ser asignado, en relación a los demás grupos que también tendrán que ser asignados. Los procesos independientes, es decir, que no formen grupos, serán agrupados y considerados como el último grupo, su cardinalidad será el número de asignaciones que lo conformen y su prioridad cero. Esto constituye lo que se denominará Prioridad de Grupo Global (**PGG**), ver Fig. 10:



*Un proceso **p** sólo puede pertenecer a un grupo a la vez.

Fig. 10. Cálculo de la **PGG** de cada grupo

Fuente: Elaboración propia.

$$PGG_i = i = 1, \dots, g = \frac{\sum FASDCO_j}{\text{Cantidad de asignaciones global del grupo } i} \quad g = \text{total de grupos en el sistema (sumatoria de grupo de los nodos); } j = n^\circ \text{ de asignaciones del grupo } i.$$

Del vector **PGG** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario global de grupo, como se muestra en Fig. 11.

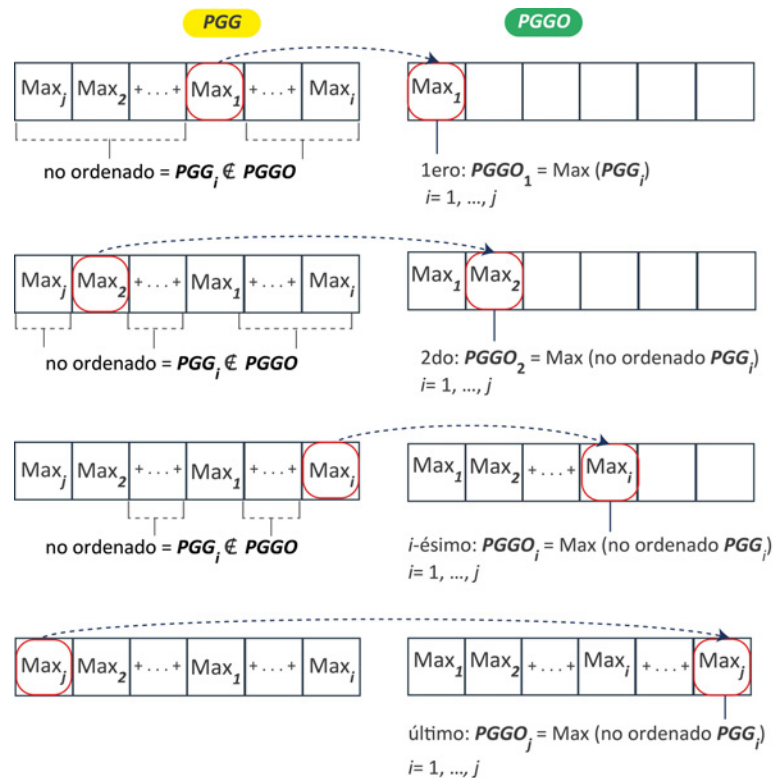


Fig. 11. Ejemplo para calcular la **PGGO** de cada grupo de procesos.
Fuente: Elaboración propia.

PGGO_i = Prioridad Global de Grupo Ordenada

g = cardinalidad de **PGG** (total de grupos en el sistema)

$$PGGO_i = \text{Max}(PGG_i \text{ no ordenado}) \quad i = 1, \dots, g$$

No ordenado = $PGG_i \notin PGGO$

Orden de prioridad global de los grupos:

$$1\text{ero: } PGGO_1 = \text{Max}(PGG_i) \quad i = 1, \dots, g$$

$$2\text{do: } PGGO_2 = \text{Max}(PGG_i \text{ no ordenado}) \quad i = 1, \dots, g$$

...

Último: $PGGO_j = \text{Max}(PGG_i \text{ no ordenado}) \quad i = 1, \dots, g$

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo
(**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDCO** y la **PGGO**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGGO** en la tabla **FASDCO**.

$cg_i =$ cardinalidad grupo(**PGGO**_{*i*}) en **FASDCO**

Ej:

$cg_1 =$ cardinalidad grupo(**PGGO**₁) en **FASDCO**

$cg_2 =$ cardinalidad grupo(**PGGO**₂) en **FASDCO**

$cg_3 =$ cardinalidad grupo(**PGGO**₃) en **FASDCO**

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

FASDOG₁ = primer elemento de la **FASDCO** para el grupo(**PGGO**₁)

...

FASDOG _{cg_1} = último elemento de la **FASDCO** para el grupo(**PGGO**₁)

FASDOG _{cg_1+1} = primer elemento de la **FASDCO** para el grupo(**PGGO**₂)

...

FASDOG _{cg_1+cg_2} = último elemento de la **FASDCO** para el grupo(**PGGO**₂)

...

...

FASDOG_{cg1+cg2+...+cgk-1+1} = primer elemento de la **FASDCO** para el grupo(**PGGO**_k)

...

FASDOG_{cg1+cg2+...+cgk} = último elemento de la **FASDCO** para el grupo(**PGGO**_k)

FASDOG_{cg1+cg2+...+cgn-1+1} = primer elemento de la **FASDCO** para el grupo(**PGGO**_n)

...

FASDOG_{cg1+cg2+...+cgn} = último elemento de la **FASDCO** para el grupo(**PGGO**_n)

Se designarán rondas de asignaciones a grupos, y para respetar la condición de exclusión mutua, habrá que establecer subrondas de asignaciones para aquellos procesos que no generen incompatibilidad en el acceso a los recursos, y otras subrondas para atender las asignaciones de recursos a procesos que no han sido atendidos en la subronda anterior, por estar en situación de incompatibilidad.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo Compatible (**FASDOGC**)

La **FASDOGC** establecerá el orden de asignación por prioridades de grupo, de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno, considerando subrondas de asignación para procesos que no generen incompatibilidad en el acceso al recurso.

Dos asignaciones se consideran incompatibles, cuando involucran al mismo recurso y a procesos distintos, lo cual, violaría el principio de exclusión mutua.

Los procesos del mismo grupo que requieran un mismo recurso, lo recibirán sucesivamente en las distintas subrondas, según la prioridad de la asignación.

Las rondas se darán a los grupos, las subrondas a las asignaciones de recursos a procesos que no generen incompatibilidad primero, y luego otras subrondas para aquellos que generaban incompatibilidad.

Se calcularán las cardinalidades de subgrupo (número de asignaciones de

recursos a procesos por grupo compatible) obtenidas de cada uno de los grupos del vector **PGGO** en la tabla **FASDCO**.

c_{sub_i} = cardinalidad de subgrupo compatible(**PGGO_i**) en **FASDCO**

Para calcular la **FASDOGC**, se tendrá en cuenta la tabla **FASDOG** y la **PGGO**:

FASDOGC_i = Max (**FASDOG_i** no ordenado compatible) $i = 1, \dots, j$

No ordenado = {a_i} de la **FASDOG** \notin **FASDOGC**

Compatible = que la asignación no involucre a un recurso que \in **FASDOGC**

La primera asignación corresponde al primer grupo de la **PGGO**, y conforma la primer ronda, luego, cada iteración de asignaciones de recursos a procesos que sean no ordenadas y compatibles, es decir, que no hayan sido cargadas (no ordenadas) en la matriz (**FASDOG**) y que sean compatibles, es decir, que no generen conflictos en el acceso a un recurso, conformaran cada subronda.

Primer subronda del primer grupo

FASDOGC₁ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO₁**)

...

FASDOGC_{cg1sub1} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO₁**)

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la primer subronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la segunda subronda.

Segunda subronda del primer grupo

FASDOGC_{cg1sub1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO₁**)

...

FASDOGC_{cg1sub1+cg1sub2} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

Se continúa este procedimiento para las **k**-subbrondas.

k subbronda del primer grupo

FASDOGC_{cg1subg1+cg1sub2+...+cg1subk-1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

...

FASDOGC_{cg1sub1+cg1sub2+...+cg1subk} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

Finalmente, se completa la ronda del primer grupo, con la última subbronda correspondiente.

Última subbronda del primer grupo

FASDOGC_{cg1sub1+cg1sub2+...+cg1subn-1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

...

FASDOGC_{cg1sub1+cg1sub2+...+cg1subn} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

Una vez asignados todos los recursos no ordenados compatibles del primer grupo, se pasará a la siguiente ronda del siguiente grupo. La cantidad de subbrondas lo determinará la cardinalidad por subbrondas calculadas anteriormente. Se repite el mismo proceso del paso anterior para cada uno de los grupos.

Primer subbronda del segundo grupo

FASDOGC_{cg1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

...

FASDOGC_{cg1+cg2sub1} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

Segunda subbronda del segundo grupo

$FASDOGC_{cg1+cg2sub1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

...

$FASDOGC_{cg1+cg2sub1+cg1sub2}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

k subbronda del segundo grupo

$FASDOGC_{cg1+cg2subg1+cg2sub2+...+cg2subk-1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

...

$FASDOGC_{cg1+cg2sub1+cg2sub2+...+cg2subk}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

Última subbronda del segundo grupo

$FASDOGC_{cg1+cg2sub1+cg2sub2+...+cg2subn-1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

...

$FASDOGC_{cg1+cg2sub1+cg2sub2+...+cg2subn}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

Se repite el mismo proceso del paso anterior para cada uno de los **k**-grupos.

Primer subbronda del grupo k

$FASDOGC_{cg1+cg2+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**_k)

...

$FASDOGC_{cg1+cg2+cgksub1}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**_k)

Segunda subbronda del grupo k

$FASDOGC_{cg1+cg2+cgksub1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**_k)

...

$FASDOGC_{cg1+cg2+cgksub1+cgksub2}$ = último elemento no ordenado compatible de la **FASDOG**

para el grupo(**PGGO_k**)

...

i-ésima subronda del grupo k

FASDOGC_{cg1+cg2+cgksub1+cgksub2+...+cgksubi-1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_k**)

...

FASDOGC_{cg1+cg2+cgksub1+cgksub2+...+cgksubi} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_k**)

Última subronda del grupo k

FASDOGC_{cg1+cg2+cgksub1+cgksub2+...+cgksubn-1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_k**)

...

FASDOGC_{cg1+cg2+cgksub1+cgksub2+...+cgksubn} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_k**)

Finalmente, se debe atender las solicitudes del último grupo y cada una de sus respectivas sub rondas calculadas con la cardinalidad de sub rondas de este grupo.

Primer subronda del grupo n

FASDOGC_{cg1+cg2+...+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_n**)

...

FASDOGC_{cg1+cg2+...+cgsub1} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_n**)

Segunda subronda del grupo n

FASDOGC_{cg1+cg2+...+cgsub1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_n**)

...

FASDOGC_{cg1+cg2+...+cgsub1+cgsub2} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_n**)

i-ésima subronda del grupo n

$FASDOGC_{cg1+cg2+\dots+cgsub1+cgsub2+\dots+cgsubi-1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_n**)

...

$FASDOGC_{cg1+cg2+\dots+cgsub1+cgsub2+\dots+cgsubi}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_n**)

Última subronda del grupo n

$FASDOGC_{cg1+cg2+\dots+cgsub1+cgsub2+\dots+cgsubn-1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_n**)

...

$FASDOGC_{cg1+cg2+\dots+cgsub1+cgsub2+\dots+cgsubn}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO_n**)

TABLA 45. Esquema general de rondas y de subrondas de cada grupo

Ronda de asignación de recursos por grupo	Subronda de asignación de recursos según compatibilidad	Proceso al que se asignará el recurso
<i>h-ésima ronda:</i> $PGGO_h = \text{Max}(PGG_h)$ $h = 1, \dots, s$	<i>primer subronda de asignación de recursos a procesos que no generan incompatibilidad</i>	$FASDOGC_1 =$ Max (FASDOG_i no ordenado compatible) $i = 1, \dots, j$... $FASDOGC_{cg1sub1} =$ Max (FASDOG_i no ordenado compatible) $i = 1, \dots, j$
	<i>g-ésima subronda de asignación de recursos a procesos que no generan incompatibilidad</i>	$FASDOGC_{cg1sub1+\dots+1} =$ Max (FASDOG_i no ordenado compatible) $i = 1, \dots, j$... $FASDOGC_{cg1sub1+\dots+cg1subi} =$ Max (FASDOG_i no ordenado compatible) $i = 1, \dots, j$
	<i>última subronda de asignación de recursos a procesos que no generan incompatibilidad</i>	$FASDOGC_{cg1sub1+\dots+cg1subn-1+1} =$ Max (FASDOG_i no ordenado compatible) $i = 1, \dots, j$... $FASDOGC_{cg1sub1+\dots+cg1subn} =$ Max (FASDOG_i no ordenado compatible) $i = 1, \dots, j$

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, ver Tabla 45.

4.5. Ejemplo

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. Todas las figuras y tablas que consignan fuente de elaboración propia, basado en La Red Martínez (2017), se transcriben en el ejemplo del escenario (E1).

La **FASDCO** se puede observar en la Tabla 44 detallada en el escenario (E2).

TABLA 46. Orden Final de asignación en la **FASDCO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.4730645	r_{12}	p_{37}	1	g_1
0.3512097	r_{11}	p_{37}	1	g_1
0.3300000	r_{21}	p_{37}	1	g_1
0.2379032	r_{32}	p_{37}	2	g_1
0.2872581	r_{33}	p_{37}	4	g_1
0.3704839	r_{31}	p_{34}	1	g_4
0.3440323	r_{22}	p_{34}	1	g_4
0.3032258	r_{32}	p_{34}	1	g_4
0.1895161	r_{24}	p_{34}	1	g_4
0.4065323	r_{33}	p_{34}	2	g_4
0.3995161	r_{12}	p_{34}	2	g_4
0.2570161	r_{13}	p_{34}	2	g_4
0.1732258	r_{23}	p_{34}	2	g_4
0.2627419	r_{21}	p_{25}	2	g_1
0.3286290	r_{13}	p_{13}	1	g_4
0.3034677	r_{31}	p_{13}	2	g_4
0.3467742	r_{33}	p_{13}	3	g_4
0.2233065	r_{22}	p_{13}	3	g_4
0.2123387	r_{11}	p_{13}	3	g_4
0.1998387	r_{21}	p_{13}	3	g_4
0.1752419	r_{32}	p_{13}	3	g_4
0.2763710	r_{12}	p_{13}	4	g_4
0.4679839	r_{33}	p_{23}	1	g_4
0.3344355	r_{12}	p_{23}	3	g_4
0.0951613	r_{24}	p_{23}	3	g_4
0.1963710	r_{31}	p_{23}	4	g_4
0.1171774	r_{32}	p_{23}	4	g_4
0.2491936	r_{23}	p_{11}	1	g_1
0.2815323	r_{11}	p_{11}	2	g_1
0.2702419	r_{22}	p_{11}	2	g_1
0.1343548	r_{24}	p_{11}	2	g_1
0.2245968	r_{12}	p_{11}	5	g_1
0.0774194	r_{21}	p_{11}	6	g_1
0.1861290	r_{13}	p_{31}	3	g_3
0.1518548	r_{31}	p_{31}	5	g_3
0.1828226	r_{33}	p_{31}	6	g_3
0.1797581	r_{22}	p_{12}	4	g_2
0.1572581	r_{21}	p_{12}	4	g_2

TABLA 46. Orden Final de asignación en la **FASDCO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.1431452	r_{11}	p_{12}	4	g_2
0.2279839	r_{33}	p_{12}	5	g_2
0.1766936	r_{12}	p_{12}	6	g_2
0.1141129	r_{31}	p_{12}	6	g_2
0.2425000	r_{31}	p_{21}	3	g_2
0.1079032	r_{23}	p_{21}	3	g_2
0.1362903	r_{13}	p_{21}	4	g_2
0.1384677	r_{22}	p_{21}	5	g_2
0.1405645	r_{33}	p_{21}	7	g_2
0.1366936	r_{12}	p_{21}	7	g_2
0.1159677	r_{21}	p_{22}	5	g_3
0.0994355	r_{22}	p_{22}	6	g_3
0.0766936	r_{31}	p_{22}	7	g_3
0.0986290	r_{33}	p_{22}	8	g_3
0.0709677	r_{23}	p_{32}	4	g_0
0.0970968	r_{13}	p_{32}	5	g_0
0.0899194	r_{11}	p_{32}	5	g_0
0.0379839	r_{12}	p_{32}	10	g_0
0.0668548	r_{32}	p_{36}	5	g_0
0.0411290	r_{24}	p_{36}	5	g_0
0.0698387	r_{13}	p_{36}	6	g_0
0.0660484	r_{11}	p_{36}	6	g_0
0.0478226	r_{31}	p_{36}	8	g_0
0.0214516	r_{21}	p_{36}	8	g_0
0.0844355	r_{12}	p_{36}	9	g_0
0.0121774	r_{22}	p_{36}	9	g_0
0.0195968	r_{33}	p_{36}	10	g_0
0.0440323	r_{23}	p_{33}	5	g_0
0.0430645	r_{21}	p_{33}	7	g_0
0.0426613	r_{11}	p_{33}	7	g_0
0.0210484	r_{32}	p_{33}	7	g_0
0.1097581	r_{12}	p_{33}	8	g_0
0.0330645	r_{22}	p_{33}	8	g_0
0.0210484	r_{13}	p_{33}	8	g_0
0.0658871	r_{33}	p_{33}	9	g_0
0.0629839	r_{24}	p_{35}	4	g_0
0.0432258	r_{32}	p_{35}	6	g_0
0.0544355	r_{22}	p_{35}	7	g_0
0.0435484	r_{13}	p_{35}	7	g_0
0.0221774	r_{31}	p_{35}	9	g_0
0.0425000	r_{33}	p_{35}	10	g_0
0.0169355	r_{12}	p_{35}	11	g_0
0.0205645	r_{23}	p_{24}	6	g_0
0.0202419	r_{24}	p_{24}	6	g_0
0.0203226	r_{11}	p_{24}	8	g_0
0.0603226	r_{12}	p_{24}	10	g_0

Fuente: Elaboración propia.

La Tabla 46 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.

Una vez que se haya completado la tabla de la **FASDCO**, con la información de grupo de cada proceso, se calcularán las Prioridades Finales Globales del Grupo (**PGG**):

$$PGG_1 = (0.4730645 + 0.3512097 + 0.33 + 0.2379032 + 0.2872581 + 0.2627419 +$$

$$0.2491936 + 0.2815323 + 0.2702419 + 0.1343548 + 0.2245968 + 0.0774194) / 12$$

$$PGG_2 = (0.1797581 + 0.1572581 + 0.1431452 + 0.2279839 + 0.1766936 + 0.1141129 + 0.2425 + 0.1079032 + 0.1362903 + 0.1384677 + 0.1405645 + 0.1366936) / 12$$

$$PGG_3 = (0.186129 + 0.1518548 + 0.1828226 + 0.1159677 + 0.0994355 + 0.0766936 + 0.098629) / 7$$

$$PGG_4 = (0.3704839 + 0.3440323 + 0.3032258 + 0.1895161 + 0.4065323 + 0.3995161 + 0.2570161 + 0.1732258 + 0.328629 + 0.3034677 + 0.3467742 + 0.2233065 + 0.2123387 + 0.1998387 + 0.1752419 + 0.276371 + 0.4679839 + 0.3344355 + 0.0951613 + 0.196371 + 0.1171774) / 21$$

$$PGG_0 = (0.0709677 + 0.0970968 + 0.0899194 + 0.0379839 + 0.0668548 + 0.041129 + 0.0698387 + 0.0660484 + 0.0478226 + 0.0214516 + 0.0844355 + 0.0121774 + 0.0195968 + 0.0440323 + 0.0430645 + 0.0426613 + 0.0210484 + 0.1097581 + 0.0330645 + 0.0210484 + 0.0658871 + 0.0629839 + 0.0432258 + 0.0544355 + 0.0435484 + 0.0221774 + 0.0425 + 0.0169355 + 0.0205645 + 0.0202419 + 0.0203226 + 0.0603226) / 32$$

Al calcular el **PGG** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 47.

TABLA 47. PGG (prioridad global final del grupo)

Prioridad Global Final	Grupo
0.264959678	g_1
0.158447580	g_2
0.130218893	g_3
0.272411675	g_4
0.000000000	g_0

Fuente: Elaboración propia.

Del vector **PGG** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario global de asignación a grupos, como se puede observar en la Tabla 48.

TABLA 48. PGGO (prioridad global final del grupo ordenada)

Prioridad Global Final Ordenada	Grupo
0.272411675	g_4
0.264959678	g_1
0.158447580	g_2
0.130218893	g_3
0.000000000	g_0

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo

(FASDOG)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDCO** y la **PGGO**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGGO** en la tabla **FASDCO**.

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

*Cálculo de las **FASDOG** para el primer grupo*

FASDOG₁ = primer elemento de la **FASDCO** para el grupo de procesos (**PGGO**₁)

...

FASDOG₂₁ = último elemento de la **FASDCO** para el grupo de procesos (**PGGO**₁)

*Cálculo de las **FASDOG** para el segundo grupo*

FASDOG₂₁₊₁ = primer elemento de la **FASDCO** para el grupo de procesos (**PGGO**₂)

...

FASDOG₂₁₊₁₂ = último elemento de la **FASDCO** para el grupo de procesos (**PGGO**₂)

*Cálculo de las **FASDOG** para el tercer grupo*

FASDOG₂₁₊₁₂₊₁ = primer elemento de la **FASDCO** para el grupo de procesos (**PGGO**₃)

...

FASDOG₂₁₊₁₂₊₁₂ = último elemento de la **FASDCO** para el grupo de procesos (**PGGO**₃)

*Cálculo de las **FASDOG** para el cuarto grupo*

FASDOG₂₁₊₁₂₊₁₂₊₁ = primer elemento de la **FASDCO** para el grupo de procesos (**PGGO**₄)

...

FASDOG₂₁₊₁₂₊₁₂₊₇ = último elemento de la **FASDCO** para el grupo de procesos (**PGGO**₄)

*Cálculo de las **FASDOG** para el quinto grupo*

FASDOG₂₁₊₁₂₊₁₂₊₇₊₁ = primer elemento de la **FASDCO** para el grupo de procesos (**PGGO**₅)

...

FASDOG₂₁₊₁₂₊₁₂₊₇₊₃₂ = último elemento de la **FASDCO** para el grupo de procesos (**PGGO**₅)

Se designarán rondas de asignaciones a grupos, y para respetar la condición de exclusión mutua, habrá que establecer subrondas de asignaciones para aquellos procesos que no generen incompatibilidad en el acceso a los recursos, y otras subrondas para atender las asignaciones de recursos a procesos que no han sido atendidos en la subronda anterior, por estar en situación de incompatibilidad.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo Compatible (**FASDOGC**)

La **FASDOGC** establecerá el orden de asignación por prioridades de grupo, de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno, considerando subrondas de asignación para procesos que no generen incompatibilidad en el acceso al recurso.

Dos asignaciones se consideran incompatibles, cuando involucran al mismo recurso y a procesos distintos, lo cual, violaría el principio de exclusión mutua.

Los procesos del mismo grupo que requieran un mismo recurso, lo recibirán sucesivamente en las distintas subrondas, según la prioridad de la asignación.

Las rondas se darán a los grupos, las subrondas a las asignaciones de recursos a procesos que no generen incompatibilidad primero, y luego otras subrondas para aquellos que generaban incompatibilidad.

Se calcularán las cardinalidades de subgrupo (número de asignaciones de recursos a procesos por grupo compatible) obtenidas de cada uno de los grupos del vector **PGGO** en la tabla **FASDCO**.

c_{sub_i} = cardinalidad de subgrupo compatible(**PGGO**_{*i*}) en **FASDCO**

Para calcular la **FASDOGC** se tendrá en cuenta la tabla **FASDOG** y la **PGGO**:

$$FASDOGC_i = \text{Max} (FASDOGi \text{ no ordenado compatible}) \quad i= 1, \dots, j$$

No ordenado = {a} de la **FASDOG** \notin **FASDOGC**

Compatible = que la asignación no involucre a un recurso que \in **FASDOGC**

La primera asignación corresponde al primer grupo de la **PGGO**, y conforma la primer ronda, luego, cada iteración de asignaciones de recursos a procesos que sean no ordenadas y compatibles, es decir, que no hayan sido cargadas (no ordenadas) en la matriz (**FASDOG**) y que sean compatibles, es decir, que no generen conflictos en el acceso a un recurso, conformaran cada subronda.

FASDOGC₁ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

...

FASDOGC_{cg1sub1} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

El primer paso es atender las asignaciones de recursos a procesos que no generan incompatibilidad. La Tabla 49 contiene los datos de la primera subronda de asignación compatible de la primera ronda de asignación, correspondiente al primer grupo (g₄).

TABLA 49. Primera subronda de asignación compatible para el primer grupo (g₄)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.3704839	r ₃₁	p ₃₄	1	g ₄
0.3440323	r ₂₂	p ₃₄	1	g ₄
0.3032258	r ₃₂	p ₃₄	1	g ₄
0.1895161	r ₂₄	p ₃₄	1	g ₄
0.4065323	r ₃₃	p ₃₄	2	g ₄
0.3995161	r ₁₂	p ₃₄	2	g ₄
0.2570161	r ₁₃	p ₃₄	2	g ₄
0.1732258	r ₂₃	p ₃₄	2	g ₄
0.2123387	r ₁₁	p ₁₃	3	g ₄
0.1998387	r ₂₁	p ₁₃	3	g ₄

Fuente: Elaboración propia.

Primer subronda del primer grupo

$FASDOGC_1$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo($PGGO_1$)

...

$FASDOGC_{10}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo($PGGO_1$)

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la primer subronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la segunda subronda.

$FASDOGC_{cg1sub1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo($PGGO_1$)

...

$FASDOGC_{cg1sub1+cg1sub2}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo($PGGO_1$)

Una vez asignados todos los recursos compatibles de la primer subronda, se pasará a la siguiente subronda, dentro de la misma ronda (**grupo**), contemplando aquellas asignaciones de recursos a procesos que no fueron atendidos en subrondas anteriores, por haber generado incompatibilidad, ver Tabla 50.

TABLA 50. Segunda subronda de asignación compatible para el primer grupo (g_4)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.3286290	r_{13}	p_{13}	1	g_4
0.3034677	r_{31}	p_{13}	2	g_4
0.3467742	r_{33}	p_{13}	3	g_4
0.2233065	r_{22}	p_{13}	3	g_4
0.1752419	r_{32}	p_{13}	3	g_4
0.2763710	r_{12}	p_{13}	4	g_4
0.0951613	r_{24}	p_{23}	3	g_4

Fuente: Elaboración propia.

Segunda subronda del primer grupo

$FASDOGC_{10+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo($PGGO_1$)

...

$FASDOGC_{10+7}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

Se continúa hasta completar todas las subrondas para atender a todas las solicitudes, ver Tabla 51.

TABLA 51. Tercer subronda de asignación compatible para el primer grupo (g_4)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.4679839	r_{33}	p_{23}	1	g_4
0.3344355	r_{12}	p_{23}	3	g_4
0.1963710	r_{31}	p_{23}	4	g_4
0.1171774	r_{32}	p_{23}	4	g_4

Fuente: Elaboración propia.

$FASDOGC_{cg1sub1+cg1sub2+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

...

$FASDOGC_{cg1sub1+cg1sub2+cg1sub3}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

Tercer subronda del primer grupo

$FASDOGC_{10+7+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

...

$FASDOGC_{10+7+4}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₁)

Se continúa hasta completar todas las iteraciones de todos los grupos.

Se debe reiterar el procedimiento pero para los siguientes grupos, atendiendo las asignaciones de recursos a procesos que no generan incompatibilidad, y luego retirando de las solicitudes de recursos las asignaciones ya hechas; para ser asignados a otros procesos que se encontraban en incompatibilidad.

La segunda asignación corresponde al segundo grupo de la **PGGO**, y conforma la segunda ronda, luego, cada iteración de asignaciones de recursos a procesos que sean no ordenadas y compatibles, es decir, que no hayan sido cargadas (no ordenadas) en la matriz (**FASDOG**) y que sean compatibles, es decir, que no generen conflictos en

el acceso a un recurso, conformarán cada subronda.

$FASDOGC_{cg1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

...

$FASDOGC_{cg1+cg2sub1}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

El primer paso es atender las asignaciones de recursos a procesos que no generan incompatibilidad. La Tabla 52 contiene los datos de la primera subronda de asignación compatible de la segunda ronda de asignación, correspondiente al segundo grupo (g_1).

TABLA 52. Primera subronda de asignación compatible para el segundo grupo (g_1)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.4730645	r_{12}	p_{37}	1	g_1
0.3512097	r_{11}	p_{37}	1	g_1
0.3300000	r_{21}	p_{37}	1	g_1
0.2379032	r_{32}	p_{37}	2	g_1
0.2872581	r_{33}	p_{37}	4	g_1
0.2491936	r_{23}	p_{11}	1	g_1
0.2702419	r_{22}	p_{11}	2	g_1
0.1343548	r_{24}	p_{11}	2	g_1

Fuente: Elaboración propia.

Primer subronda del segundo grupo

$FASDOGC_{21+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

...

$FASDOGC_{21+8}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la primer subronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la segunda subronda.

$FASDOGC_{cg1+cg2sub1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

...

FASDOGC_{cg1+cg2sub1+cg2sub2} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

Una vez asignados todos los recursos compatibles de la primer subronda, se pasará a la siguiente subronda, dentro de la misma ronda (**grupo**), contemplando aquellas asignaciones de recursos a procesos que no fueron atendidos en subrondas anteriores, por haber generado incompatibilidad, ver Tabla 53.

TABLA 53. Segunda subronda de asignación compatible para el primer grupo (g₁)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.2627419	r ₂₁	p ₂₅	2	g ₁
0.2815323	r ₁₁	p ₁₁	2	g ₁
0.2245968	r ₁₂	p ₁₁	5	g ₁

Fuente: Elaboración propia.

Segunda subronda del segundo grupo

FASDOGC₂₁₊₈₊₁ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

...

FASDOGC₂₁₊₈₊₃ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

Se continúa hasta completar todas las subrondas para atender a todas las solicitudes, ver Tabla 54.

TABLA 54. Tercer subronda de asignación compatible para el primer grupo (g₁)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0774194	r ₂₁	p ₁₁	6	g ₁

Fuente: Elaboración propia.

FASDOGC_{cg1+cg2sub1+ cg2sub2+ cg2sub3} = único elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

FASDOGC₂₁₊₈₊₃₊₁ = único elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

Tercer subronda del segundo grupo

FASDOGC₂₁₊₈₊₃₊₁ = único elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₂)

Se continúa hasta completar todas las iteraciones de todos los grupos.

Se debe reiterar el procedimiento pero para los siguientes grupos, atendiendo las asignaciones de recursos a procesos que no generan incompatibilidad, y luego retirando de las solicitudes de recursos las asignaciones ya hechas; para ser asignados a otros procesos que se encontraban en incompatibilidad.

La tercer asignación corresponde al tercer grupo de la **PGGO**, y conforma la tercer ronda, luego, cada iteración de asignaciones de recursos a procesos que sean no ordenadas y compatibles, es decir, que no hayan sido cargadas (no ordenadas) en la matriz (**FASDOG**) y que sean compatibles, es decir, que no generen conflictos en el acceso a un recurso, conformarán cada subronda.

FASDOGC_{cg1+cg2+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₃)

...

FASDOGC_{cg1+cg2+cg3sub1} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₃)

El primer paso es atender las asignaciones de recursos a procesos que no generan incompatibilidad. La Tabla 55 contiene los datos de la primera subronda de asignación compatible de la segunda ronda de asignación, correspondiente al segundo grupo (g_2).

TABLA 55. Primera subronda de asignación compatible para el tercer grupo (g_2)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.1797581	r_{22}	p_{12}	4	g_2
0.1572581	r_{21}	p_{12}	4	g_2
0.1431452	r_{11}	p_{12}	4	g_2
0.2279839	r_{33}	p_{12}	5	g_2
0.1766936	r_{12}	p_{12}	6	g_2
0.1141129	r_{31}	p_{12}	6	g_2
0.1079032	r_{23}	p_{21}	3	g_2
0.1362903	r_{13}	p_{21}	4	g_2

Fuente: Elaboración propia.

Primer subronda del tercer grupo

FASDOGC₂₁₊₁₂₊₁= primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₃)

...

$FASDOGC_{21+12+8}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₃)

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la primer subronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la segunda subronda.

$FASDOGC_{cg1+cg2+cg3sub1+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₃)

...

$FASDOGC_{cg1+cg2+cg3sub1+cg3sub2}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₃)

Una vez asignados todos los recursos compatibles de la primer subronda, se pasará a la siguiente subronda, dentro de la misma ronda (**grupo**), contemplando aquellas asignaciones de recursos a procesos que no fueron atendidos en subrondas anteriores, por haber generado incompatibilidad, ver Tabla 56.

TABLA 56. Segunda subronda de asignación compatible para el tercer grupo (*g*₂)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.2425000	<i>r</i> ₃₁	<i>p</i> ₂₁	3	<i>g</i> ₂
0.1384677	<i>r</i> ₂₂	<i>p</i> ₂₁	5	<i>g</i> ₂
0.1405645	<i>r</i> ₃₃	<i>p</i> ₂₁	7	<i>g</i> ₂
0.1366936	<i>r</i> ₁₂	<i>p</i> ₂₁	7	<i>g</i> ₂

Fuente: Elaboración propia.

Segunda subronda del segundo grupo

$FASDOGC_{21+12+8+1}$ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₃)

...

$FASDOGC_{21+12+8+4}$ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₃)

Se continúa hasta completar todas las iteraciones de todos los grupos.

Se debe reiterar el procedimiento pero para los siguientes grupos, atendiendo las asignaciones de recursos a procesos que no generan incompatibilidad, y luego retirando

de las solicitudes de recursos las asignaciones ya hechas; para ser asignados a otros procesos que se encontraban en incompatibilidad.

La cuarta asignación corresponde al cuarto grupo de la **PGGO**, y conforma la cuarta ronda, luego, cada iteración de asignaciones de recursos a procesos que sean no ordenadas y compatibles, es decir, que no hayan sido cargadas (no ordenadas) en la matriz (**FASDOG**) y que sean compatibles, es decir, que no generen conflictos en el acceso a un recurso, conformaran cada subronda.

FASDOGC_{cg1+cg2+cg3+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₄)

...

FASDOGC_{cg1+cg2+cg3+cg4sub1} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₄)

El primer paso es atender las asignaciones de recursos a procesos que no generan incompatibilidad. La Tabla 57 contiene los datos de la primera subronda de asignación compatible de la segunda ronda de asignación, correspondiente al cuarto grupo (g_3).

TABLA 57. Primera subronda de asignación compatible para el cuarto grupo (g_3)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.1861290	r_{13}	p_{31}	3	g_3
0.1518548	r_{31}	p_{31}	5	g_3
0.1828226	r_{33}	p_{31}	6	g_3
0.1159677	r_{21}	p_{22}	5	g_3
0.0994355	r_{22}	p_{22}	6	g_3

Fuente: Elaboración propia.

Primer subronda del cuarto grupo

FASDOGC₂₁₊₁₂₊₁₂₊₁ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₄)

...

FASDOGC₂₁₊₁₂₊₁₂₊₅ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₄)

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de

asignaciones ya realizadas en la primer subronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la segunda subronda.

FASDOGC_{cg1+cg2+cg3+cg4sub1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₄)

...

FASDOGC_{cg1+cg2+cg3+cg4sub1+cg4sub2} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₄)

Una vez asignados todos los recursos compatibles de la primer subronda, se pasará a la siguiente subronda, dentro de la misma ronda (**grupo**), contemplando aquellas asignaciones de recursos a procesos que no fueron atendidos en subrondas anteriores, por haber generado incompatibilidad, ver Tabla 58.

Tabla 58. Segunda subronda de asignación compatible para el cuarto grupo (*g₃*)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0766936	<i>r</i> ₃₁	<i>p</i> ₂₂	7	<i>g</i> ₃
0.0986290	<i>r</i> ₃₃	<i>p</i> ₂₂	8	<i>g</i> ₃

Fuente: Elaboración propia.

Segunda subronda del cuarto grupo

FASDOGC₂₁₊₁₂₊₁₂₊₅₊₁ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₄)

...

FASDOGC₂₁₊₁₂₊₁₂₊₅₊₂ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₄)

Se continúa hasta completar todas las iteraciones del último grupo, conformado por los procesos que no pertenecían a ningún grupo, los cuales fueron asignados al grupo 0.

Se debe reiterar el procedimiento pero para el último grupo, atendiendo las asignaciones de recursos a procesos que no generan incompatibilidad, y luego retirando de las solicitudes de recursos las asignaciones ya hechas; para ser asignados a otros procesos que se encontraban en incompatibilidad.

La quinta asignación corresponde al quinto grupo de la **PGGO**, y conforma la quinta ronda, luego, cada iteración de asignaciones de recursos a procesos que sean no ordenadas y compatibles, es decir, que no hayan sido cargadas (no ordenadas) en la matriz (**FASDOG**) y que sean compatibles, es decir, que no generen conflictos en el acceso a un recurso, conformaran cada subronda.

FASDOGC_{cg1+cg2+cg3+cg4+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

...

FASDOGC_{cg1+cg2+cg3+cg4+cg5sub1} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

El primer paso es atender las asignaciones de recursos a procesos que no generan incompatibilidad. La Tabla 59 contiene los datos de la primera subronda de asignación compatible de la segunda ronda de asignación, correspondiente al quinto grupo (g_0).

TABLA 59. Primera subronda de asignación compatible para el quinto grupo (g_0)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0709677	r_{23}	p_{32}	4	g_0
0.0970968	r_{13}	p_{32}	5	g_0
0.0899194	r_{11}	p_{32}	5	g_0
0.0379839	r_{12}	p_{32}	10	g_0
0.0668548	r_{32}	p_{36}	5	g_0
0.0411290	r_{24}	p_{36}	5	g_0
0.0478226	r_{31}	p_{36}	8	g_0
0.0214516	r_{21}	p_{36}	8	g_0
0.0121774	r_{22}	p_{36}	9	g_0
0.0195968	r_{33}	p_{36}	10	g_0

Fuente: Elaboración propia.

Primer subronda del quinto grupo

FASDOGC₂₁₊₁₂₊₁₂₊₇₊₁ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

...

FASDOGC₂₁₊₁₂₊₁₂₊₇₊₁₀ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de

asignaciones ya realizadas en la primer subronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la segunda subronda.

FASDOGC_{cg1+cg2+cg3+cg4+cg5sub1+1} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

...

FASDOGC_{cg1+cg2+cg3+cg4+cg5sub1+cg5sub2} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

Una vez asignados todos los recursos compatibles de la primer subronda, se pasará a la siguiente subronda, dentro de la misma ronda (**grupo**), contemplando aquellas asignaciones de recursos a procesos que no fueron atendidos en subrondas anteriores, por haber generado incompatibilidad, ver Tabla 60.

TABLA 60. Segunda subronda de asignación compatible para el quinto grupo (*g₀*)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0698387	<i>r</i> ₁₃	<i>p</i> ₃₆	6	<i>g</i> ₀
0.0660484	<i>r</i> ₁₁	<i>p</i> ₃₆	6	<i>g</i> ₀
0.0844355	<i>r</i> ₁₂	<i>p</i> ₃₆	9	<i>g</i> ₀
0.0440323	<i>r</i> ₂₃	<i>p</i> ₃₃	5	<i>g</i> ₀
0.0430645	<i>r</i> ₂₁	<i>p</i> ₃₃	7	<i>g</i> ₀
0.0210484	<i>r</i> ₃₂	<i>p</i> ₃₃	7	<i>g</i> ₀
0.0330645	<i>r</i> ₂₂	<i>p</i> ₃₃	8	<i>g</i> ₀
0.0658871	<i>r</i> ₃₃	<i>p</i> ₃₃	9	<i>g</i> ₀
0.0629839	<i>r</i> ₂₄	<i>p</i> ₃₅	4	<i>g</i> ₀
0.0221774	<i>r</i> ₃₁	<i>p</i> ₃₅	9	<i>g</i> ₀

Fuente: Elaboración propia.

Segunda subronda del segundo grupo

FASDOGC₂₁₊₁₂₊₁₂₊₇₊₁₀₊₁ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

...

FASDOGC₂₁₊₁₂₊₁₂₊₇₊₁₀₊₁₀ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

Se continúa hasta completar todas las subrondas para atender a todas las solicitudes, ver Tabla 61.

TABLA 61. Tercer subbronda de asignación compatible para el quinto grupo (g_0)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0426613	r_{11}	p_{33}	7	g_0
0.1097581	r_{12}	p_{33}	8	g_0
0.0210484	r_{13}	p_{33}	8	g_0
0.0432258	r_{32}	p_{35}	6	g_0
0.0544355	r_{22}	p_{35}	7	g_0
0.0425000	r_{33}	p_{35}	10	g_0
0.0205645	r_{23}	p_{24}	6	g_0
0.0202419	r_{24}	p_{24}	6	g_0

Fuente: Elaboración propia.

FASDOGC _{$cg1+cg2+cg3+cg4+cg5sub1+cg5sub2+1$} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

...

FASDOGC _{$cg1+cg2+cg3+cg4+cg5sub1+cg5sub2+cg5sub3$} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

Tercer subbronda del quinto grupo

FASDOGC _{$21+12+12+7+10+10+1$} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

...

FASDOGC _{$21+12+12+7+10+10+8$} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

Se continúa hasta completar todas la última subbronda para atender a todas las solicitudes, ver Tabla 62.

TABLA 62. Cuarta subbronda de asignación compatible para el quinto grupo (g_0)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0435484	r_{13}	p_{35}	7	g_0
0.0169355	r_{12}	p_{35}	11	g_0
0.0203226	r_{11}	p_{24}	8	g_0
0.0603226	r_{12}	p_{24}	10	g_0

Fuente: Elaboración propia.

FASDOGC _{$cg1+cg2+cg3+cg4+cg5sub1+cg5sub2+cg5sub3+1$} = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

...

FASDOGC_{cg1+cg2+cg3+cg4+cg5sub1+cg5sub2+cg5sub3+cg5sub4} = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

Cuarta subbronda del quinto grupo

FASDOGC₂₁₊₁₂₊₁₂₊₇₊₁₀₊₁₀₊₈₊₁ = primer elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

...

FASDOGC₂₁₊₁₂₊₁₂₊₇₊₁₀₊₁₀₊₈₊₄ = último elemento no ordenado compatible de la **FASDOG** para el grupo(**PGGO**₅)

La Tabla 63 muestra el orden final de todas las asignaciones de recursos para cada proceso ordenada por grupo compatible (**FASDOGC**).

TABLA 63. Orden Final de asignación en la FASDOGC

Prioridad	Recursos	Procesos	Rondas	Grupo
0.3704839	r ₃₁	p ₃₄	1	g ₄
0.3440323	r ₂₂	p ₃₄	1	g ₄
0.3032258	r ₃₂	p ₃₄	1	g ₄
0.1895161	r ₂₄	p ₃₄	1	g ₄
0.4065323	r ₃₃	p ₃₄	2	g ₄
0.3995161	r ₁₂	p ₃₄	2	g ₄
0.2570161	r ₁₃	p ₃₄	2	g ₄
0.1732258	r ₂₃	p ₃₄	2	g ₄
0.3286290	r ₁₃	p ₁₃	1	g ₄
0.3034677	r ₃₁	p ₁₃	2	g ₄
0.3467742	r ₃₃	p ₁₃	3	g ₄
0.2233065	r ₂₂	p ₁₃	3	g ₄
0.2123387	r ₁₁	p ₁₃	3	g ₄
0.1998387	r ₂₁	p ₁₃	3	g ₄
0.1752419	r ₃₂	p ₁₃	3	g ₄
0.2763710	r ₁₂	p ₁₃	4	g ₄
0.4679839	r ₃₃	p ₂₃	1	g ₄
0.3344355	r ₁₂	p ₂₃	3	g ₄
0.0951613	r ₂₄	p ₂₃	3	g ₄
0.1963710	r ₃₁	p ₂₃	4	g ₄
0.1171774	r ₃₂	p ₂₃	4	g ₄
0.4730645	r ₁₂	p ₃₇	1	g ₁
0.3512097	r ₁₁	p ₃₇	1	g ₁
0.3300000	r ₂₁	p ₃₇	1	g ₁
0.2379032	r ₃₂	p ₃₇	2	g ₁
0.2872581	r ₃₃	p ₃₇	4	g ₁
0.2627419	r ₂₁	p ₂₅	2	g ₁
0.2491936	r ₂₃	p ₁₁	1	g ₁
0.2815323	r ₁₁	p ₁₁	2	g ₁
0.2702419	r ₂₂	p ₁₁	2	g ₁
0.1343548	r ₂₄	p ₁₁	2	g ₁
0.2245968	r ₁₂	p ₁₁	5	g ₁
0.0774194	r ₂₁	p ₁₁	6	g ₁
0.1797581	r ₂₂	p ₁₂	4	g ₂
0.1572581	r ₂₁	p ₁₂	4	g ₂
0.1431452	r ₁₁	p ₁₂	4	g ₂
0.2279839	r ₃₃	p ₁₂	5	g ₂
0.1766936	r ₁₂	p ₁₂	6	g ₂
0.1141129	r ₃₁	p ₁₂	6	g ₂
0.2425000	r ₃₁	p ₂₁	3	g ₂
0.1079032	r ₂₃	p ₂₁	3	g ₂

TABLA 63. Orden Final de asignación en la FASDOGC

Prioridad	Recursos	Procesos	Rondas	Grupo
0.1362903	r_{13}	p_{21}	4	g_2
0.1384677	r_{22}	p_{21}	5	g_2
0.1405645	r_{33}	p_{21}	7	g_2
0.1366936	r_{12}	p_{21}	7	g_2
0.1861290	r_{13}	p_{31}	3	g_3
0.1518548	r_{31}	p_{31}	5	g_3
0.1828226	r_{33}	p_{31}	6	g_3
0.1159677	r_{21}	p_{22}	5	g_3
0.0994355	r_{22}	p_{22}	6	g_3
0.0766936	r_{31}	p_{22}	7	g_3
0.0986290	r_{33}	p_{22}	8	g_3
0.0709677	r_{23}	p_{32}	4	g_0
0.0970968	r_{13}	p_{32}	5	g_0
0.0899194	r_{11}	p_{32}	5	g_0
0.0379839	r_{12}	p_{32}	10	g_0
0.0668548	r_{32}	p_{36}	5	g_0
0.0411290	r_{24}	p_{36}	5	g_0
0.0698387	r_{13}	p_{36}	6	g_0
0.0660484	r_{11}	p_{36}	6	g_0
0.0478226	r_{31}	p_{36}	8	g_0
0.0214516	r_{21}	p_{36}	8	g_0
0.0844355	r_{12}	p_{36}	9	g_0
0.0121774	r_{22}	p_{36}	9	g_0
0.0195968	r_{33}	p_{36}	10	g_0
0.0440323	r_{23}	p_{33}	5	g_0
0.0430645	r_{21}	p_{33}	7	g_0
0.0426613	r_{11}	p_{33}	7	g_0
0.0210484	r_{32}	p_{33}	7	g_0
0.1097581	r_{12}	p_{33}	8	g_0
0.0330645	r_{22}	p_{33}	8	g_0
0.0210484	r_{13}	p_{33}	8	g_0
0.0658871	r_{33}	p_{33}	9	g_0
0.0629839	r_{24}	p_{35}	4	g_0
0.0432258	r_{32}	p_{35}	6	g_0
0.0544355	r_{22}	p_{35}	7	g_0
0.0435484	r_{13}	p_{35}	7	g_0
0.0221774	r_{31}	p_{35}	9	g_0
0.0425000	r_{33}	p_{35}	10	g_0
0.0169355	r_{12}	p_{35}	11	g_0
0.0205645	r_{23}	p_{24}	6	g_0
0.0202419	r_{24}	p_{24}	6	g_0
0.0203226	r_{11}	p_{24}	8	g_0
0.0603226	r_{12}	p_{24}	10	g_0

Fuente: Elaboración propia.

De esta manera, se respondió a todas las solicitudes de recursos de todos los procesos de cada grupo, 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.

4.6. Evaluación

El modelo propuesto logra establecer un consenso que permite a los grupos de procesos acceder a todos sus recursos de manera secuencial y que éstos no pueden

ser removidos hasta que el mismo grupo de procesos que los mantiene los libera. El orden de asignación será determinado por la prioridad promedio general de todas las asignaciones de cada grupo. 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 grupos de 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 grupo de procesos tiene sobre todos los recursos de todas sus asignaciones en las diferentes rondas **PGGO**, pero para la asignación global final del grupo, establece subrondas compatibles de asignación, dado que en un mismo grupo puede haber procesos que requieran un mismo recurso, por lo tanto, los procesos de mayor prioridad serán asignados en una primer subronda y los que generan incompatibilidad se asignarán en una siguiente subronda.

Es decir, la elección del grupo de procesos al que se le otorgarán recursos, se establece con el promedio global de prioridades en todas las asignaciones, pero el orden en el que se deben realizar esas asignaciones, se establece por medio de subrondas compatibles de asignación.

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 a los proceso p_{ij} (cuyo primer subíndice representa al nodo donde se encuentra y el segundo subíndice al propio número de proceso) en cada ronda. Aunque para los recursos/procesos los subíndices son iguales, no necesariamente sean los mismos recursos/procesos, sino que pueden representar a distintos recurso/proceso

que se asigna varias veces en las distintas rondas, pero siempre al mismo grupo g_h .

Para las cardinalidades, “ cg ” corresponde a la identificación del grupo, y “ sub ” a la subronda de asignación que pertenece. Se debe continuar con el mismo procedimiento para los siguientes grupos en la **PGGO**, ver Fig. 12.

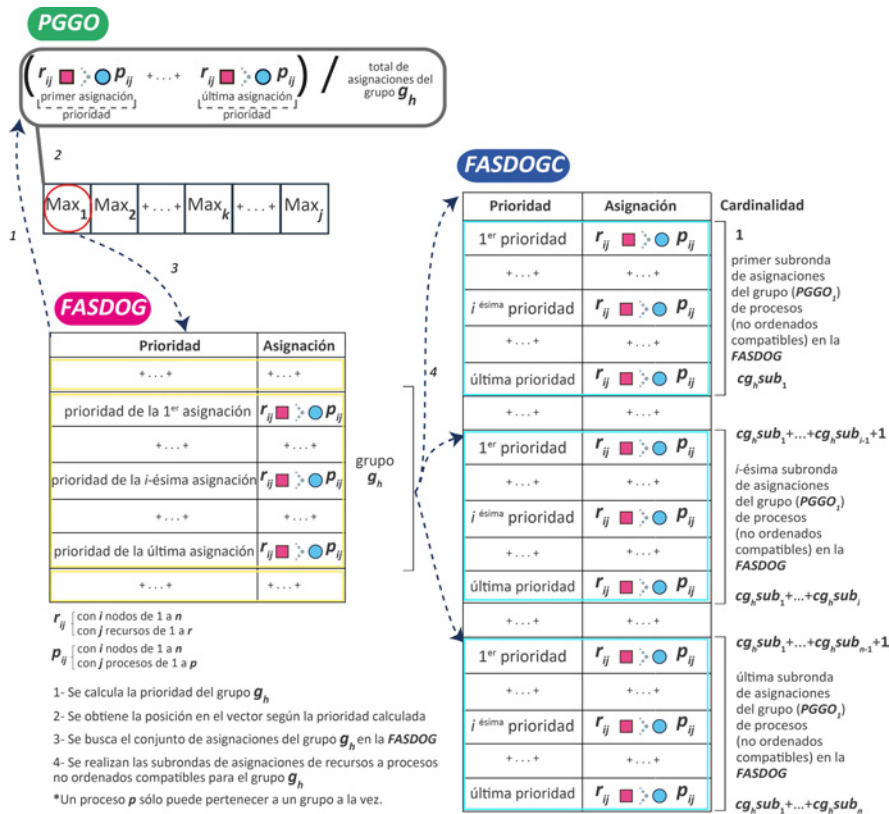


Fig. 12. Listado final del grupo de mayor prioridad en la **PGGO**

Fuente: Elaboración propia.

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

4.7. Discusiones y comentarios

Como los recursos son asignados por subrondas de asignaciones compatibles, no hay conflicto en cuanto a que varios procesos requieran el mismo recurso.

Como los procesos se ejecutan en diferentes procesadores utilizando todos sus recursos, no hay conflicto en la ejecución de varios procesos en el mismo procesador.

En este escenario, no se tiene en cuenta la cantidad de tiempo que cada proceso

utilizará en un procesador de un nodo en particular. Tampoco lo es la cantidad de tiempo en que cada recurso será asignado a un proceso de cada grupo en particular.

Los procesos independientes, es decir, que no forman grupos, son agrupados y considerados como el último grupo, su cardinalidad es el número de procesos que lo conforman y su prioridad cero, favoreciendo a los procesos que pertenecen a grupos.

En el siguiente capítulo, se desarrolla una variante del método propuesto considerando otro operador de agregación para grupos de procesos que no requieren requisitos de consenso estrictos.

El contenido de este capítulo sirvió de base para las siguiente publicación:

- Agostini F., La Red Martínez D. L., (2019). "*Allocation of shared resources*". 14th Iberian Conference on Information Systems and Technologies - CISTI 2019. ISBN N° 978-989-98434-9-3, pp. 1-6. Universidad de Coimbra; Coimbra, Portugal.

Capítulo V - Operador de agregación definido para el escenario 4 (E4)

5.1. Introducción

Trabajos previos

En (La Red Martínez, 2017) y (La Red Martínez et al. 2018) se desarrollan operadores de agregación para la asignación de recursos en sistemas distribuidos.

5.2. Hipótesis y objetivos

Que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua **constituyendo grupos de procesos** (los procesos independientes son considerados grupos unitarios); los procesos no requieren sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y **sin exigencias estrictas de consenso para lograr el acceso** (no se requiere un consenso para asignar de manera consecutiva los recursos solicitados por un procesos, es decir, que iniciada la secuencia de asignación de recursos al grupo, la misma puede ser interrumpida para asignar recursos a otro proceso).

5.3. Propuestas de solución

Se presentará una variante de un método innovador para la gestión de recursos compartidos en sistemas distribuidos, basado en (La Red Martínez, 2017) y (La Red Martínez et al. 2018), donde se desarrolla un operador de agregación, para asignar recursos en sistemas distribuidos, pero en este caso, estableciendo un orden de asignación por grupo de procesos en cada instancia. Las premisas, las estructuras de datos y el operador mencionado en el segundo capítulo se utilizan como punto de partida para crear un nuevo operador en el escenario descrito a continuación.

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 ser capaces de formar grupos de procesos (los procesos independientes se consideran grupos unitarios y con prioridad grupal cero), en

tercer lugar, los procesos no deben requerir sincronización (es decir, estar activos en sus respectivos procesadores al mismo tiempo) y no deben requerir requisitos de consenso para acceder a los recursos.

Se trata de grupos de procesos distribuidos en nodos de procesos que acceden a recursos críticos compartidos en la modalidad de exclusión mutua distribuida, debiendo decidirse, ante la demanda de recursos por parte de los procesos, cuáles serán las prioridades para asignar los recursos a los grupos de procesos que los requieren (sólo intervendrán como alternativas de asignación a los procesos aquellos recursos disponibles, es decir, no asignados aún a determinados procesos):

- El permiso de acceso a los recursos compartidos propios de un nodo no dependerá sólo de si los nodos los están utilizando o no, sino del valor de agregación de las preferencias (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas).
- Las opiniones (prioridades) de los distintos nodos respecto de otorgar el acceso a los recursos compartidos (alternativas) dependerá de la consideración del valor de variables que representen el estado de cada uno de los distintos nodos. Cada nodo deberá expresar sus prioridades para la asignación de los distintos recursos compartidos respecto de los requerimientos de recursos de cada proceso.
- Prioridad de grupo, que se entiende por priorizar al grupo de procesos que resulte con mayor prioridad promedio en cada asignación, según las tablas de asignaciones.

5.4. Descripción del operador de agregación

El operador propuesto consta de las siguientes etapas:

1. Cálculo de la carga computacional actual de los nodos.
2. Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a las mismas.

3. 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).

4. 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.

5. Cálculo de las prioridades de los grupos de procesos para acceder a los recursos compartidos disponibles y determinación del orden en que se asignarán los recursos y a qué grupo de procesos serán asignados cada uno de ellos.

En la Fig. 13 se muestra una lista de los pasos necesarios para obtener las prioridades globales definitivas para asignar los recursos (***Función de Asignación para Sistemas Distribuidos Ordenada- FASDO***) hasta obtener el orden o prioridad de asignación de los recursos y el proceso al que se asigna cada recurso ordenada por grupo (***Función de Asignación para Sistemas Distribuidos Ordenada por Grupo - FASDOG***).

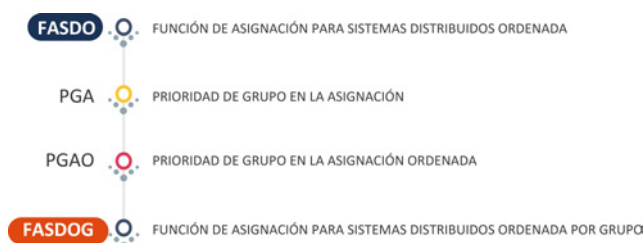


Fig. 13. Pasos para obtener desde la ***FASDO*** a la ***FASDOG***
Fuente: Elaboración propia.

Una vez calculada la ***FASDO***, se establecerá la **prioridad grupal** y el orden en el que se asignarán los recursos a procesos en los grupos. Para ellos se tendrá en cuenta todos los procesos y recursos vinculados en cada instancia. La cardinalidad de grupo se medirá por la cantidad de procesos que hay en cada grupo, en cada instancia.

Prioridad del grupo en la asignación

La sumatoria de todas las prioridades finales (***FASDO***) de cada relación proceso

y recurso por grupo, dividido la cardinalidad (números de asignaciones de cada grupo, en cada instancias) del grupo, indicará la prioridad que tendrá dicho grupo para ser asignado, en relación a los demás grupos que también tendrán que ser asignados. Los procesos independientes, es decir, que no formen grupos, serán agrupados y considerados como el último grupo, su cardinalidad será el número de asignaciones que lo conformen y su prioridad cero. Esto constituye lo que se denominará Prioridad de Grupo en la Asignación (**PGA**):

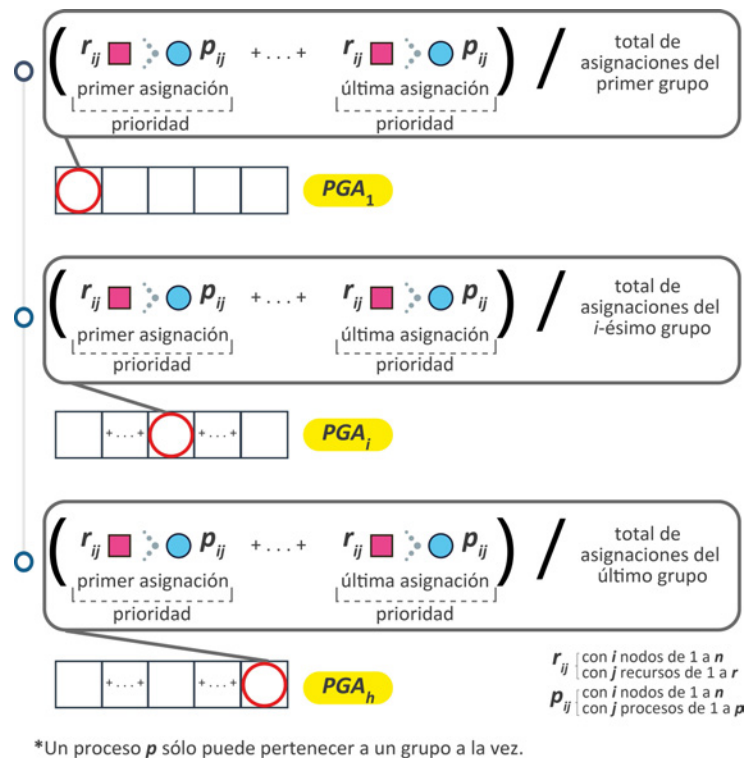


Fig. 14. Cálculo de la **PGA** de cada grupo en la asignación
Fuente: Elaboración propia.

$$PGA_i = i = 1, \dots, g = \frac{\sum FASDOj}{\text{Cardinalidad del grupo } i/\text{asignación}} \quad g = \text{total de grupos en la}$$

asignación; $j = n^\circ$ de asignaciones del grupo i en la ronda.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario del grupo en la ronda de asignación, como se muestra en Fig. 14.

PGA_O= Prioridad del Grupo en la Asignación Ordenada

g = cardinalidad de **PGA** (total de grupos en la asignación)

PGA_O _{i} = Max (**PGA_i**; no ordenado) $i = 1, \dots, g$

No ordenado = $PGA_i \notin PGO$

Orden de prioridad global de los grupos en la asignación, ver Fig. 15:

1ero: $PGAO_1 = \text{Max} (PGA_i) \quad i = 1, \dots, g$

2do: $PGAO_2 = \text{Max} (PGA_i \text{ no ordenado}) \quad i = 1, \dots, g$

...

Último: $PGAO_j = \text{Max} (PGA_i \text{ no ordenado}) \quad i = 1, \dots, g$

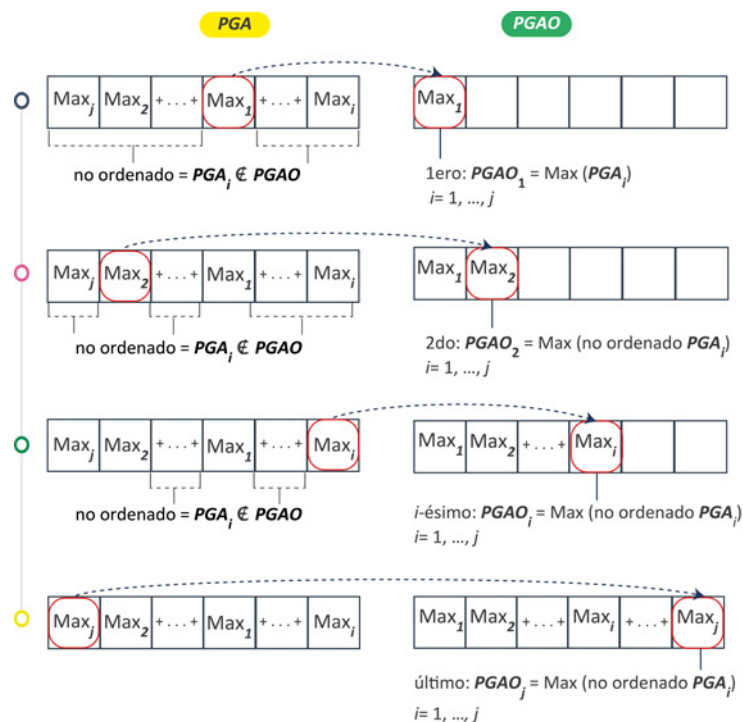


Fig. 15. Ejemplo para calcular la **PGAO** de cada grupo de procesos
Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDCO** y la **PGAO**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGAO** en la tabla **FASDO**, en la ronda.

$cg_j =$ cardinalidad grupo($PGAO_j$) en **FASDO**

Ej:

cg_1 = cardinalidad grupo(**PGA0₁**) en **FASDO**

cg_2 = cardinalidad grupo(**PGA0₂**) en **FASDO**

cg_3 = cardinalidad grupo(**PGA0₃**) en **FASDO**

Luego habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla **FASDO** de cada uno de los grupos del vector **PGA0**. El total de elementos por cada grupo, estará determinado por la cardinalidad calculada en el paso anterior.

FASDOG₁ = primer elemento de la **FASDO** para el grupo(**PGA0₁**)

...

FASDOG_{cg1} = último elemento de la **FASDO** para el grupo(**PGA0₁**)

FASDOG_{cg1+1} = primer elemento de la **FASDO** para el grupo(**PGA0₂**)

...

FASDOG_{cg1+cg2} = último elemento de la **FASDO** para el grupo(**PGA0₂**)

...

...

FASDOG_{cg1+cg2+...+cgk-1+1} = primer elemento de la **FASDO** para el grupo(**PGA0_k**)

...

FASDOG_{cg1+cg2+...+cgk} = último elemento de la **FASDO** para el grupo(**PGA0_k**)

FASDOG_{cg1+cg2+...+cgn-1+1} = primer elemento de la **FASDO** para el grupo(**PGA0_n**)

...

FASDOG_{cg1+cg2+...+cgn} = último elemento de la **FASDO** para el grupo(**PGA0_n**)

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la primer ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros grupos de procesos en la siguiente ronda, ver Tabla 64.

TABLA 64. Esquema general de rondas y de subrondas de cada grupo

Ronda de asignación	Grupos en la asignación	Proceso del grupo al que se asignará el recurso en la asignación
g-ésima ronda:	$PGAO_1 =$ $Max (PGA_h)$ $h= 1, \dots, s$	$FASDOG_1 = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$ $FASDOG_2 = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$... $FASDOG_n = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$
	$PGAO_e =$ $Max (PGA_h \text{ no ordenado})$ $h= 1, \dots, s$	$FASDOG_1 = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$ $FASDOG_2 = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$... $FASDOG_n = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$
	$PGAO_n =$ $Max (PGA_h \text{ no ordenado})$ $h= 1, \dots, s$	$FASDOG_1 = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$ $FASDOG_2 = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$... $FASDOG_n = Max (FASDO_i \text{ no ordenado})$ $i= 1, \dots, j$

Fuente: Elaboración propia.

Dado que se utiliza la tabla **FASDO** por cada asignación, se mantiene la condición de exclusión mutua.

No hay procesos del mismo grupo que requieran un mismo recurso, si se puede dar, que un proceso pueda requerir más de un recurso.

Se continúa hasta completar todas las rondas de asignación.

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 requisitos establecidos para este escenario.

5.5. Ejemplo

El escenario que se presenta a continuación, parte de la asignación ordenada de la primera iteración correspondiente al escenario antes mencionado. Todas las figuras y tablas que consignan fuente de elaboración propia, basado en La Red Martínez (2017), se transcriben en el ejemplo del escenario (E1).

La tabla **FASDO** se obtiene de la Tabla 22 del escenario anterior (E2).

La Tabla 65 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la asignación.

TABLA 65. Orden de asignación en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.4730645	r_{12}	p_{37}	1	g_1
0.4679839	r_{33}	p_{23}	1	g_4
0.3704839	r_{31}	p_{34}	1	g_4
0.3512097	r_{11}	p_{37}	1	g_1
0.3440323	r_{22}	p_{34}	1	g_4
0.3300000	r_{21}	p_{37}	1	g_1
0.3286290	r_{13}	p_{13}	1	g_4
0.3032258	r_{32}	p_{34}	1	g_4
0.2491936	r_{23}	p_{11}	1	g_1
0.1895161	r_{24}	p_{34}	1	g_4

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asginación (**PGA**):

$$PGA_1 = (0.4730645 + 0.3512097 + 0.33 + 0.2491936) / 4$$

$$PGA_2 = (0.4679839 + 0.3704839 + 0.3440323 + 0.3286290 + 0.3032258 + 0.1895161) / 6$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 66.

TABLA 66. **PGA** (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.35086695	g_1
0.33397850	g_4

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos por asignación, como se puede observar en la Tabla 67.

TABLA 67. **PGA**O (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.35086695	g_1
0.33397850	g_4

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo

(FASDOG)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGA0**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGA0** en la tabla **FASDO**, de la *primer asignación*.

Luego habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla **FASDO** de cada uno de los grupos del vector **PGA0**. El total de elementos por cada grupo, estará determinado por la cardinalidad calculada en el paso anterior.

Cálculo de las FASDOG para el primer grupo

FASDOG₁ = primer elemento de la **FASDO** para el grupo (**PGA0₁**)

...

FASDOG₄ = último elemento de la **FASDO** para el grupo (**PGA0₁**)

Cálculo de las FASDOG para el segundo grupo

FASDOG₄₊₁ = primer elemento de la **FASDO** para el grupo (**PGA0₂**)

...

FASDOG₄₊₆ = último elemento de la **FASDO** para el grupo (**PGA0₂**)

La Tabla 68 muestra el orden final de asignación para la primera iteración de la **FASDOG**.

TABLA 68. Orden de asignación de cada grupo en la **FASDOG** (primer iteración)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.4730645	<i>r</i> ₁₂	<i>p</i> ₃₇	1	<i>g</i> ₁
0.3512097	<i>r</i> ₁₁	<i>p</i> ₃₇	1	<i>g</i> ₁
0.3300000	<i>r</i> ₂₁	<i>p</i> ₃₇	1	<i>g</i> ₁
0.2491936	<i>r</i> ₂₃	<i>p</i> ₁₁	1	<i>g</i> ₁
0.4679839	<i>r</i> ₃₃	<i>p</i> ₂₃	1	<i>g</i> ₄
0.3704839	<i>r</i> ₃₁	<i>p</i> ₃₄	1	<i>g</i> ₄
0.3440323	<i>r</i> ₂₂	<i>p</i> ₃₄	1	<i>g</i> ₄

TABLA 68. Orden de asignación de cada grupo en la **FASDOG** (primer iteración)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.3286290	r_{13}	p_{13}	1	g_4
0.3032258	r_{32}	p_{34}	1	g_4
0.1895161	r_{24}	p_{34}	1	g_4

Fuente: Elaboración propia.

Se continúa hasta completar todas las iteraciones del siguiente grupo.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la primer ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la segunda ronda.

La Tabla 69 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la segunda asignación.

TABLA 69. Orden de asignación por proceso en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.2379032	r_{32}	p_{37}	2	g_1
0.4065323	r_{33}	p_{34}	2	g_4
0.3995161	r_{12}	p_{34}	2	g_4
0.2570161	r_{13}	p_{34}	2	g_4
0.1732258	r_{23}	p_{34}	2	g_4
0.2627419	r_{21}	p_{25}	2	g_1
0.3034677	r_{31}	p_{13}	2	g_4
0.2815323	r_{11}	p_{11}	2	g_1
0.2702419	r_{22}	p_{11}	2	g_1
0.1343548	r_{24}	p_{11}	2	g_1

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asignación (**PGA**):

$$PGA_1 = (0.2379032 + 0.2627419 + 0.2815323 + 0.2702419 + 0.1343548) / 5$$

$$PGA_2 = (0.4065323 + 0.3995161 + 0.2570161 + 0.1732258 + 0.3034677) / 5$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 70.

TABLA 70. *PGA* (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.23735482	g_1
0.30795160	g_4

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 71.

TABLA 71. *PGAO* (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.30795160	g_4
0.23735482	g_1

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGAO**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGAO** en la tabla **FASDO**, de la segunda asignación.

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

Cálculo de las FASDOG para el primer grupo

$FASDOG_1$ = primer elemento de la **FASDO** para el grupo ($PGAO_1$)

...

$FASDOG_5$ = último elemento de la **FASDO** para el grupo ($PGAO_1$)

Cálculo de las FASDOG para el segundo grupo

FASDOG₅₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO**₂)

...

FASDOG₅₊₅ = último elemento de la **FASDO** para el grupo (**PGAO**₂)

La Tabla 72 muestra el orden final de asignación para la segunda iteración de la **FASDOG**.

Tabla 72. Orden de asignación de grupo en la **FASDOG** (segunda asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.4065323	<i>r</i> ₃₃	<i>p</i> ₃₄	2	<i>g</i> ₄
0.3995161	<i>r</i> ₁₂	<i>p</i> ₃₄	2	<i>g</i> ₄
0.2570161	<i>r</i> ₁₃	<i>p</i> ₃₄	2	<i>g</i> ₄
0.1732258	<i>r</i> ₂₃	<i>p</i> ₃₄	2	<i>g</i> ₄
0.3034677	<i>r</i> ₃₁	<i>p</i> ₁₃	2	<i>g</i> ₄
0.2379032	<i>r</i> ₃₂	<i>p</i> ₃₇	2	<i>g</i> ₁
0.2627419	<i>r</i> ₂₁	<i>p</i> ₂₅	2	<i>g</i> ₁
0.2815323	<i>r</i> ₁₁	<i>p</i> ₁₁	2	<i>g</i> ₁
0.2702419	<i>r</i> ₂₂	<i>p</i> ₁₁	2	<i>g</i> ₁
0.1343548	<i>r</i> ₂₄	<i>p</i> ₁₁	2	<i>g</i> ₁

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la segunda ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la tercer ronda.

La Tabla 73 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por grupo de procesos con mayor prioridad en la tercera asignación.

TABLA 73 . Orden de asignación por proceso en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.3467742	<i>r</i> ₃₃	<i>p</i> ₁₃	3	<i>g</i> ₄
0.2233065	<i>r</i> ₂₂	<i>p</i> ₁₃	3	<i>g</i> ₄
0.2123387	<i>r</i> ₁₁	<i>p</i> ₁₃	3	<i>g</i> ₄
0.1998387	<i>r</i> ₂₁	<i>p</i> ₁₃	3	<i>g</i> ₄
0.1752419	<i>r</i> ₃₂	<i>p</i> ₁₃	3	<i>g</i> ₄
0.3344355	<i>r</i> ₁₂	<i>p</i> ₂₃	3	<i>g</i> ₄
0.0951613	<i>r</i> ₂₄	<i>p</i> ₂₃	3	<i>g</i> ₄
0.1861290	<i>r</i> ₁₃	<i>p</i> ₃₁	3	<i>g</i> ₃
0.2425000	<i>r</i> ₃₁	<i>p</i> ₂₁	3	<i>g</i> ₂
0.1079032	<i>r</i> ₂₃	<i>p</i> ₂₁	3	<i>g</i> ₂

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asignación (**PGA**):

$$PGA_1 = (0.3467742 + 0.2233065 + 0.2123387 + 0.1998387 + 0.1752419 + 0.3344355 + 0.0951613) / 7$$

$$PGA_2 = (0.1861290) / 1$$

$$PGA_3 = (0.2425000 + 0.1079032) / 2$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 74.

TABLA 74 . PGA (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.226728114	g_4
0.186129000	g_3
0.175201600	g_2

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 75.

TABLA 75. PGA (prioridad de grupo en la asignación ordenada)

Prioridad Global Final	Grupo
0.226728114	g_4
0.186129000	g_3
0.175201600	g_2

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGA**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGA** en la tabla **FASDO**, de la tercer asignación.

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

Cálculo de las FASDOG para el primer grupo

FASDOG₁ = primer elemento de la **FASDO** para el grupo (**PGAO₁**)

...

FASDOG₇ = último elemento de la **FASDO** para el grupo (**PGAO₁**)

Cálculo de las FASDOG para el segundo grupo

FASDOG₇₊₁ = único elemento de la **FASDO** para el grupo (**PGAO₂**)

Cálculo de las FASDOG para el tercer grupo

FASDOG₇₊₁₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO₃**)

...

FASDOG₇₊₁₊₂ = último elemento de la **FASDO** para el grupo (**PGAO₃**)

La Tabla 76 muestra el orden final de asignación para la tercera iteración de la **FASDOG**.

TABLA 76. Orden de asignación de grupo en la **FASDOG** (tercer asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.3467742	<i>r</i> ₃₃	<i>p</i> ₁₃	3	<i>g</i> ₄
0.2233065	<i>r</i> ₂₂	<i>p</i> ₁₃	3	<i>g</i> ₄
0.2123387	<i>r</i> ₁₁	<i>p</i> ₁₃	3	<i>g</i> ₄
0.1998387	<i>r</i> ₂₁	<i>p</i> ₁₃	3	<i>g</i> ₄
0.1752419	<i>r</i> ₃₂	<i>p</i> ₁₃	3	<i>g</i> ₄
0.3344355	<i>r</i> ₁₂	<i>p</i> ₂₃	3	<i>g</i> ₄
0.0951613	<i>r</i> ₂₄	<i>p</i> ₂₃	3	<i>g</i> ₄
0.1861290	<i>r</i> ₁₃	<i>p</i> ₃₁	3	<i>g</i> ₃
0.2425000	<i>r</i> ₃₁	<i>p</i> ₂₁	3	<i>g</i> ₂
0.1079032	<i>r</i> ₂₃	<i>p</i> ₂₁	3	<i>g</i> ₂

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la tercer ronda; debe tenerse en cuenta que los recursos

asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la cuarta ronda.

La Tabla 77 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la cuarta asignación.

TABLA 77. Orden de asignación en la FASDO

Prioridad	Recursos	Procesos	Rondas	Grupo
0.2872581	r_{33}	p_{37}	4	g_1
0.2763710	r_{12}	p_{13}	4	g_4
0.1963710	r_{31}	p_{23}	4	g_4
0.1171774	r_{32}	p_{23}	4	g_4
0.1797581	r_{22}	p_{12}	4	g_2
0.1572581	r_{21}	p_{12}	4	g_2
0.1431452	r_{11}	p_{12}	4	g_2
0.1362903	r_{13}	p_{21}	4	g_2
0.0709677	r_{23}	p_{32}	4	g_0
0.0629839	r_{24}	p_{35}	4	g_0

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asignación (**PGA**):

$$PGA_1 = (0.2872581) / 1$$

$$PGA_2 = (0.2763710 + 0.1963710 + 0.1171774) / 3$$

$$PGA_3 = (0.1797581 + 0.1572581 + 0.1431452 + 0.1362903) / 4$$

$$PGA_4 = (0.0709677 + 0.0629839) / 2$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 78.

TABLA 78. PGA (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.287258100	g_1
0.196639800	g_4
0.154112925	g_2
0.066975800	g_0

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 79.

TABLA 79. P_{GAO} (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.287258100	g_1
0.196639800	g_4
0.154112925	g_2
0.066975800	g_0

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **P_{GAO}**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **P_{GAO}** en la tabla **FASDO**, de la cuarta asignación.

Luego habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla **FASDO** de cada uno de los grupos del vector **P_{GAO}**. El total de elementos por cada grupo, estará determinado por la cardinalidad calculada en el paso anterior.

Cálculo de las FASDOG para el primer grupo

FASDOG₁ = único elemento de la **FASDO** para el grupo (**P_{GAO}₁**)

Cálculo de las FASDOG para el segundo grupo

FASDOG₁₊₁ = primer elemento de la **FASDO** para el grupo (**P_{GAO}₂**)

...

FASDOG₁₊₃ = último elemento de la **FASDO** para el grupo (**P_{GAO}₂**)

Cálculo de las FASDOG para el tercer grupo

FASDOG₁₊₃₊₁ = primer elemento de la **FASDO** para el grupo (**P_{GAO}₃**)

...

FASDOG₁₊₃₊₄ = último elemento de la **FASDO** para el grupo (**P_{GAO}₃**)

Cálculo de las FASDOG para el cuarto grupo

FASDOG₁₊₃₊₄₊₁ = primer elemento de la **FASDO** para el grupo (**PGA0₄**)

...

FASDOG₁₊₃₊₄₊₂ = último elemento de la **FASDO** para el grupo (**PGA0₄**)

La Tabla 80 muestra el orden final de asignación para la cuarta iteración de la **FASDOG**.

TABLA 80. Orden de asignación de grupo en la **FASDOG** (cuarta asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.2872581	<i>r</i> ₃₃	<i>p</i> ₃₇	4	<i>g</i> ₁
0.2763710	<i>r</i> ₁₂	<i>p</i> ₁₃	4	<i>g</i> ₄
0.1963710	<i>r</i> ₃₁	<i>p</i> ₂₃	4	<i>g</i> ₄
0.1171774	<i>r</i> ₃₂	<i>p</i> ₂₃	4	<i>g</i> ₄
0.1797581	<i>r</i> ₂₂	<i>p</i> ₁₂	4	<i>g</i> ₂
0.1572581	<i>r</i> ₂₁	<i>p</i> ₁₂	4	<i>g</i> ₂
0.1431452	<i>r</i> ₁₁	<i>p</i> ₁₂	4	<i>g</i> ₂
0.1362903	<i>r</i> ₁₃	<i>p</i> ₂₁	4	<i>g</i> ₂
0.0709677	<i>r</i> ₂₃	<i>p</i> ₃₂	4	<i>g</i> ₀
0.0629839	<i>r</i> ₂₄	<i>p</i> ₃₅	4	<i>g</i> ₀

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la cuarta ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la quinta ronda.

La Tabla 81 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la quinta asignación.

TABLA 81. Orden de asignación en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.2245968	<i>r</i> ₁₂	<i>p</i> ₁₁	5	<i>g</i> ₁
0.1518548	<i>r</i> ₃₁	<i>p</i> ₃₁	5	<i>g</i> ₃
0.2279839	<i>r</i> ₃₃	<i>p</i> ₁₂	5	<i>g</i> ₂
0.1384677	<i>r</i> ₂₂	<i>p</i> ₂₁	5	<i>g</i> ₂
0.1159677	<i>r</i> ₂₁	<i>p</i> ₂₂	5	<i>g</i> ₃
0.0970968	<i>r</i> ₁₃	<i>p</i> ₃₂	5	<i>g</i> ₀
0.0899194	<i>r</i> ₁₁	<i>p</i> ₃₂	5	<i>g</i> ₀
0.0668548	<i>r</i> ₃₂	<i>p</i> ₃₆	5	<i>g</i> ₀

TABLA 81. Orden de asignación en la FASDO

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0411290	r_{24}	p_{36}	5	g_0
0.0440323	r_{23}	p_{33}	5	g_0

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asginación (**PGA**):

$$PGA_1 = (0.2245968) / 1$$

$$PGA_2 = (0.1518548 + 0.1159677) / 2$$

$$PGA_3 = (0.2279839 + 0.1384677) / 2$$

$$PGA_4 = (0.0970968 + 0.0899194 + 0.0668548 + 0.0411290 + 0.0440323) / 5$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 82.

TABLA 82. PGA (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.22459680	g_1
0.13391125	g_3
0.18322580	g_2
0.06780646	g_0

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 83.

TABLA 83. PGAO (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.22459680	g_1
0.18322580	g_2
0.13391125	g_3
0.06780646	g_0

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGAO**.

Se calcularán las cardinalidades (número de asignaciones de recursos a

procesos) obtenidas de cada uno de los grupos del vector **PGAO** en la tabla **FASDO**, de la quinta asignación.

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

*Cálculo de las **FASDOG** para el primer grupo*

FASDOG₁ = único elemento de la **FASDO** para el grupo (**PGAO₁**)

*Cálculo de las **FASDOG** para el segundo grupo*

FASDOG₁₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO₂**)

...

FASDOG₁₊₂ = último elemento de la **FASDO** para el grupo (**PGAO₂**)

*Cálculo de las **FASDOG** para el tercer grupo*

FASDOG₁₊₂₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO₃**)

...

FASDOG₁₊₂₊₂ = último elemento de la **FASDO** para el grupo (**PGAO₃**)

*Cálculo de las **FASDOG** para el cuarto grupo*

FASDOG₁₊₂₊₂₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO₄**)

...

FASDOG₁₊₂₊₂₊₅ = último elemento de la **FASDO** para el grupo (**PGAO₄**)

La Tabla 84 muestra el orden final de asignación para la quinta iteración de la **FASDOG**.

TABLA 84. Orden de asignación de grupo en la **FASDOG** (quinta asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.2245968	r_{12}	p_{11}	5	g_1
0.2279839	r_{33}	p_{12}	5	g_2
0.1384677	r_{22}	p_{21}	5	g_2
0.1518548	r_{31}	p_{31}	5	g_3
0.1159677	r_{21}	p_{22}	5	g_3
0.0970968	r_{13}	p_{32}	5	g_0

TABLA 84. Orden de asignación de grupo en la **FASDOG** (quinta asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0899194	r_{11}	p_{32}	5	g_0
0.0668548	r_{32}	p_{36}	5	g_0
0.0411290	r_{24}	p_{36}	5	g_0
0.0440323	r_{23}	p_{33}	5	g_0

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la quinta ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la sexta ronda.

La Tabla 85 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la sexta asignación.

TABLA 85. Orden de asignación en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0774194	r_{21}	p_{11}	6	g_1
0.1828226	r_{33}	p_{31}	6	g_3
0.1766936	r_{12}	p_{12}	6	g_2
0.1141129	r_{31}	p_{12}	6	g_2
0.0994355	r_{22}	p_{22}	6	g_3
0.0698387	r_{13}	p_{36}	6	g_0
0.0660484	r_{11}	p_{36}	6	g_0
0.0432258	r_{32}	p_{35}	6	g_0
0.0205645	r_{23}	p_{24}	6	g_0
0.0202419	r_{24}	p_{24}	6	g_0

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asginación (**PGA**):

$$PGA_1 = (0.0774194) / 1$$

$$PGA_2 = (0.1828226 + 0.0994355) / 2$$

$$PGA_3 = (0.1766936 + 0.1141129) / 2$$

$$PGA_4 = (0.0698387 + 0.0660484 + 0.0432258 + 0.0205645 + 0.0202419) / 5$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 86.

TABLA 86. *PGA* (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.07741940	g_1
0.14112905	g_3
0.14540325	g_2
0.04398386	g_0

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 87.

TABLA 87 . *PGA*O (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.14540325	g_2
0.14112905	g_3
0.0774194	g_1
0.04398386	g_0

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGA**O.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGA**O en la tabla **FASDO**, de la sexta asignación.

Luego habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla **FASDO** de cada uno de los grupos del vector **PGA**O. El total de elementos por cada grupo, estará determinado por la cardinalidad calculada en el paso anterior.

*Cálculo de las **FASDOG** para el primer grupo*

FASDOG₁ = único elemento de la **FASDO** para el grupo (**PGA**O₁)

*Cálculo de las **FASDOG** para el segundo grupo*

FASDOG₁₊₁ = primer elemento de la **FASDO** para el grupo (**PGA**O₂)

...

FASDOG₁₊₂ = último elemento de la **FASDO** para el grupo (**PGAO**₂)

Cálculo de las FASDOG para el tercer grupo

FASDOG₁₊₂₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO**₃)

...

FASDOG₁₊₂₊₂ = último elemento de la **FASDO** para el grupo (**PGAO**₃)

Cálculo de las FASDOG para el cuarto grupo

FASDOG₁₊₂₊₂₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO**₄)

...

FASDOG₁₊₂₊₂₊₅ = último elemento de la **FASDO** para el grupo (**PGAO**₄)

La Tabla 88 muestra el orden final de asignación para la sexta iteración de la

FASDOG.

TABLA 88. Orden de asignación de grupo en la **FASDOG** (sexta asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.1766936	<i>r</i> ₁₂	<i>p</i> ₁₂	6	<i>g</i> ₂
0.1141129	<i>r</i> ₃₁	<i>p</i> ₁₂	6	<i>g</i> ₂
0.1828226	<i>r</i> ₃₃	<i>p</i> ₃₁	6	<i>g</i> ₃
0.0994355	<i>r</i> ₂₂	<i>p</i> ₂₂	6	<i>g</i> ₃
0.0774194	<i>r</i> ₂₁	<i>p</i> ₁₁	6	<i>g</i> ₁
0.0698387	<i>r</i> ₁₃	<i>p</i> ₃₆	6	<i>g</i> ₀
0.0660484	<i>r</i> ₁₁	<i>p</i> ₃₆	6	<i>g</i> ₀
0.0432258	<i>r</i> ₃₂	<i>p</i> ₃₅	6	<i>g</i> ₀
0.0205645	<i>r</i> ₂₃	<i>p</i> ₂₄	6	<i>g</i> ₀
0.0202419	<i>r</i> ₂₄	<i>p</i> ₂₄	6	<i>g</i> ₀

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la sexta ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la séptima ronda.

La Tabla 89 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la séptima asignación.

TABLA 89. Orden de asignación en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.1405645	<i>r</i> ₃₃	<i>p</i> ₂₁	7	<i>g</i> ₂

TABLA 89. Orden de asignación en la FASDO

Prioridad	Recursos	Procesos	Rondas	Grupo
0.1366936	r_{12}	p_{21}	7	g_2
0.0766936	r_{31}	p_{22}	7	g_3
0.0430645	r_{21}	p_{33}	7	g_0
0.0426613	r_{11}	p_{33}	7	g_0
0.0210484	r_{32}	p_{33}	7	g_0
0.0544355	r_{22}	p_{35}	7	g_0
0.0435484	r_{13}	p_{35}	7	g_0

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asignación (**PGA**):

$$PGA_1 = (0.1405645 + 0.1366936) / 2$$

$$PGA_2 = (0.0766936) / 1$$

$$PGA_3 = (0.0430645 + 0.0426613 + 0.0210484 + 0.0544355 + 0.0435484) / 5$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 90.

TABLA 90. PGA (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.13862905	g_2
0.07669360	g_3
0.04095162	g_0

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 91.

TABLA 91. PGAO (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.13862905	g_2
0.07669360	g_3
0.04095162	g_0

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGAO**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGAO** en la tabla **FASDO**, de la séptima asignación.

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

Cálculo de las FASDOG para el primer grupo

FASDOG₁ = primer elemento de la **FASDO** para el grupo (**PGAO₁**)

...

FASDOG₂= último elemento de la **FASDO** para el grupo (**PGAO₁**)

Cálculo de las FASDOG para el segundo grupo

FASDOG₂₊₁ = único elemento de la **FASDO** para el grupo (**PGAO₂**)

Cálculo de las FASDOG para el tercer grupo

FASDOG₂₊₁₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO₃**)

...

FASDOG₂₊₁₊₅ = último elemento de la **FASDO** para el grupo (**PGAO₃**)

La Tabla 92 muestra el orden final de asignación para la séptima iteración de la **FASDOG**.

TABLA 92. Orden de asignación de grupo en la **FASDOG** (séptima asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.1405645	<i>r</i> ₃₃	<i>p</i> ₂₁	7	<i>g</i> ₂
0.1366936	<i>r</i> ₁₂	<i>p</i> ₂₁	7	<i>g</i> ₂
0.0766936	<i>r</i> ₃₁	<i>p</i> ₂₂	7	<i>g</i> ₃
0.0430645	<i>r</i> ₂₁	<i>p</i> ₃₃	7	<i>g</i> ₀
0.0426613	<i>r</i> ₁₁	<i>p</i> ₃₃	7	<i>g</i> ₀
0.0210484	<i>r</i> ₃₂	<i>p</i> ₃₃	7	<i>g</i> ₀
0.0544355	<i>r</i> ₂₂	<i>p</i> ₃₅	7	<i>g</i> ₀
0.0435484	<i>r</i> ₁₃	<i>p</i> ₃₅	7	<i>g</i> ₀

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la séptima ronda; debe tenerse en cuenta que los

recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la octava ronda.

La Tabla 93 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la octava asignación.

TABLA 93. Orden de asignación en la FASDO

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0986290	r_{33}	p_{22}	8	g_3
0.0478226	r_{31}	p_{36}	8	g_0
0.0214516	r_{21}	p_{36}	8	g_0
0.1097581	r_{12}	p_{33}	8	g_0
0.0330645	r_{22}	p_{33}	8	g_0
0.0210484	r_{13}	p_{33}	8	g_0
0.0203226	r_{11}	p_{24}	8	g_0

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asginación (**PGA**):

$$PGA_1 = (0.0986290) / 1$$

$$PGA_2 = (0.0478226 + 0.0214516 + 0.1097581 + 0.0330645 + 0.0210484 + 0.0203226)$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 94.

TABLA 94. PGA (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.09862900	g_3
0.04224463	g_0

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 95.

TABLA 95. PGAO (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.09862900	g_3
0.04224463	g_0

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el orden de asignación por prioridades de grupo de

los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGAO**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGAO** en la tabla **FASDO**, de la octava asignación.

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

Cálculo de las FASDOG para el primer grupo

FASDOG₁ = único elemento de la **FASDO** para el grupo (**PGAO₁**)

Cálculo de las FASDOG para el segundo grupo

FASDOG₁₊₁ = primer elemento de la **FASDO** para el grupo (**PGAO₂**)

...

FASDOG₁₊₆ = último elemento de la **FASDO** para el grupo (**PGAO₂**)

La Tabla 96 muestra el orden final de asignación para la octava iteración de la **FASDOG**.

TABLA 96. Orden de asignación de grupo en la **FASDOG** (octava asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0986290	r_{33}	p_{22}	8	g_3
0.0478226	r_{31}	p_{36}	8	g_0
0.0214516	r_{21}	p_{36}	8	g_0
0.1097581	r_{12}	p_{33}	8	g_0
0.0330645	r_{22}	p_{33}	8	g_0
0.0210484	r_{13}	p_{33}	8	g_0
0.0203226	r_{11}	p_{24}	8	g_0

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la octava ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la novena ronda.

La Tabla 97 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la novena asignación.

TABLA 97. Orden de asignación en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0844355	r_{12}	p_{36}	9	g_0
0.0121774	r_{22}	p_{36}	9	g_0
0.0658871	r_{33}	p_{33}	9	g_0
0.0221774	r_{31}	p_{35}	9	g_0

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asignación (**PGA**):

$$PGA_1 = (0.0844355 + 0.0121774 + 0.0658871 + 0.0221774) / 4$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 98.

TABLA 98. **PGA** (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.04616935	g_3

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 99.

TABLA 99. **PGA0** (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.04616935	g_3

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGA0**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGA0** en la tabla **FASDO**, de la novena asignación.

Luego habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla **FASDO** de cada uno de los grupos del vector **PGA0**. El total de elementos por cada grupo, estará determinado por la cardinalidad calculada en el paso anterior.

Cálculo de las FASDOG para el único grupo

FASDOG₁ = primer elemento de la **FASDO** para el grupo (**PGA0₁**)

...

FASDOG₄= último elemento de la **FASDO** para el grupo (**PGA0₁**)

La Tabla 100 muestra el orden final de asignación para la novena iteración de la **FASDOG**.

TABLA 100. Orden de asignación de grupo en la **FASDOG** (novena asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0844355	r_{12}	p_{36}	9	g_0
0.0121774	r_{22}	p_{36}	9	g_0
0.0658871	r_{33}	p_{33}	9	g_0
0.0221774	r_{31}	p_{35}	9	g_0

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la novena ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la décima ronda.

La Tabla 101 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la décima asignación.

TABLA 101. Orden de asignación en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0379839	r_{12}	p_{32}	10	g_0
0.0195968	r_{33}	p_{36}	10	g_0
0.0425000	r_{33}	p_{35}	10	g_0
0.0603226	r_{12}	p_{24}	10	g_0

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asginación (**PGA**):

$$PGA_1 = (0.0379839 + 0.0195968 + 0.0425000 + 0.0603226) / 4$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 102.

TABLA 102. PGA (prioridad de grupo en la asignación).

Prioridad Global Final	Grupo
0.04010083	g_0

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 103.

TABLA 103. PGAO (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.04010083	g_0

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo (**FASDOG**)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGAO**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGAO** en la tabla **FASDO**, de la décima asignación.

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

Cálculo de las FASDOG para el único grupo

FASDOG₁ = primer elemento de la **FASDO** para el grupo (**PGAO₁**)

...

FASDOG₄ = último elemento de la **FASDO** para el grupo (**PGAO₁**)

La Tabla 104 muestra el orden final de asignación para la décima iteración de la

FASDOG.

TABLA 104. Orden de asignación de grupo en la **FASDOG** (décima asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0379839	r_{12}	p_{32}	10	g_0
0.0195968	r_{33}	p_{36}	10	g_0
0.0425000	r_{33}	p_{35}	10	g_0
0.0603226	r_{12}	p_{24}	10	g_0

Fuente: Elaboración propia.

El siguiente paso es repetir el procedimiento, pero eliminando las solicitudes de asignaciones ya realizadas en la décima ronda; debe tenerse en cuenta que los recursos asignados estarán disponibles una vez liberados por los procesos y, por lo tanto, pueden asignarse a otros procesos en la undécima ronda.

La Tabla 105 muestra el orden de las asignaciones de recursos para cada proceso, ordenado por proceso con mayor prioridad en la undécima asignación.

TABLA 105. Orden de asignación en la **FASDO**

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0169355	r_{12}	p_{35}	11	g_0

Fuente: Elaboración propia.

Una vez que se haya completado la tabla de la **FASDO**, con la información de grupo de cada proceso, se calcularán las Prioridad del Grupo en la Asignación (**PGA**):

$$PGA_1 = (0.0169355) / 1$$

Al calcular el **PGA** para todos los grupos, se obtendrá un vector como se muestra en la Tabla 106.

TABLA 106. **PGA** (prioridad de grupo en la asignación)

Prioridad Global Final	Grupo
0.0169355	g_0

Fuente: Elaboración propia.

Del vector **PGA** se debe ordenar sus elementos de mayor a menor para obtener el orden prioritario de asignación a grupos, como se puede observar en la Tabla 107.

TABLA 107 . **PGA**O (prioridad de grupo en la asignación ordenada)

Prioridad Global Final Ordenada	Grupo
0.0169355	g_0

Fuente: Elaboración propia.

Función de Asignación para Sistemas Distribuidos Ordenada por Grupo

(FASDOG)

La **FASDOG** establecerá el **orden de asignación por prioridades de grupo** de los procesos para acceder a sus recursos y se establecerá el orden en el que se asignará cada uno. Para ello, se tendrá en cuenta la tabla **FASDO** y la **PGA0**.

Se calcularán las cardinalidades (número de asignaciones de recursos a procesos) obtenidas de cada uno de los grupos del vector **PGA0** en la tabla **FASDO**, de la undécima asignación.

Luego habrá que obtener cada una de las asignaciones de recursos a procesos en la tabla **FASDO** de cada uno de los grupos del vector **PGA0**. El total de elementos por cada grupo, estará determinado por la cardinalidad calculada en el paso anterior.

Cálculo de las FASDOG para el único grupo

FASDOG₁ = único elemento de la **FASDO** para el grupo (**PGA0₁**)

La Tabla 108 muestra el orden final de asignación para la undécima iteración de la **FASDOG**.

TABLA 108. Orden de asignación de grupo en la **FASDOG** (undécima asignación)

Prioridad	Recursos	Procesos	Rondas	Grupo
0.0169355	r_{12}	p_{35}	11	g_0

Fuente: Elaboración propia.

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

5.6. Evaluación

El modelo propuesto logra establecer un orden que permite a los grupos de procesos acceder a todos sus recursos de manera secuencial y que éstos no pueden ser removidos hasta que el grupo de procesos que los mantiene los libere. El orden de asignación será determinado por la prioridad promedio general de cada grupo en cada

asignación. 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 grupos de 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.

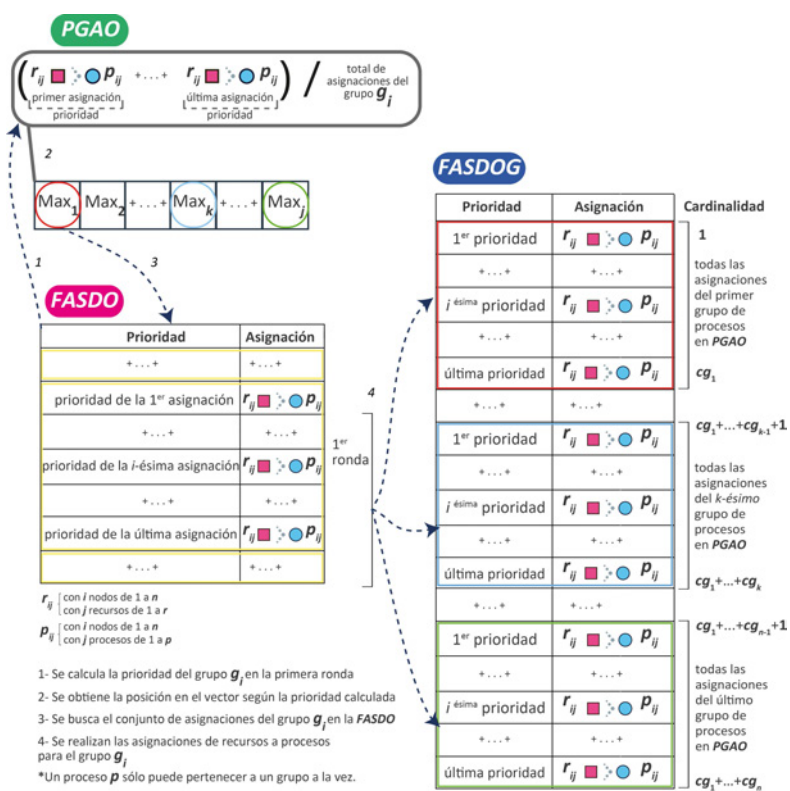


Fig. 16. Cálculo desde la FASDO a la FASDOG en la primera ronda.
Fuente: Elaboración propia.

Este capítulo considera que el grupo de procesos al cual se le otorgan los recursos, se determina con el promedio global de prioridades en todas las asignaciones en la ronda PGO, pero el orden en el que se deben realizar esas asignaciones, respeta el de la tabla FASDO, para cada grupo de procesos.

La representación de los recursos 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 a los proceso p_{ij} (cuyo primer subíndice representa al nodo donde se encuentra y el segundo subíndice al propio número de proceso) en cada ronda. Aunque para los recursos/procesos los subíndices son iguales, no necesariamente sean los mismos recursos/procesos, sino que pueden representar a distintos recurso/proceso que se asigna varias veces en las distintas rondas. Cabe destacar que los procesos deben pertenecer a un sólo grupo, ver Fig. 16.

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

5.7. Discusiones y comentarios

Como los recursos son asignados por rondas de asignaciones, no hay conflicto en cuanto a que varios procesos requieran el mismo recurso.

En este escenario, no se tiene en cuenta la cantidad de tiempo que cada proceso utilizará en un procesador de un nodo en particular. Tampoco lo es la cantidad de tiempo en que cada recurso será asignado a un proceso de cada grupo en particular.

Los procesos independientes, es decir, que no forman grupos, son agrupados y considerados como el último grupo, su cardinalidad es el número de procesos que lo conforman y su prioridad cero, favoreciendo a los procesos que pertenecen a grupos.

En el siguiente capítulo, se desarrolla el modelo de decisión y el operador de agregación general para los distintos escenarios.

El contenido de este capítulo sirvió de base para la siguiente publicación:

- Agostini F., Fornerón Martínez J. T., La Red Martínez D. L., (2019). “*Nuevo operador de agregación para grupos de procesos*”. 16ª Conferencia Ibero Americana. Lisboa, Portugal. (Aceptado).

Capítulo VI - Modelos de decisión y operadores de agregación

6.1. Introducción a los modelos de decisión

Un Modelo de Decisión permite definir los objetivos claramente, se basa en la recolección de datos y su relevancia, estudia las posibles alternativas y evalúa sus consecuencias. La tarea consiste en seleccionar la mejor alternativa para lograr la mejor solución.

6.2. Modelo de decisión propuesto

El objetivo principal del modelo propuesto es considerar el entorno de ejecución distribuida de procesos, que podrán estar agrupados, con distintas exigencias de consenso respecto del acceso a recursos compartidos, para seleccionar el método de agregación más adecuado a cada escenario posible, para generar la secuencia de asignación de los recursos a los procesos que los solicitan, respetando la exclusión mutua en el acceso a dichos recursos.

El modelo considera la información proveniente de la caracterización de los procesos (si los procesos pertenecen a grupos o son independientes, y si la aplicación a la que pertenecen requiere algún tipo de consenso respecto del acceso a los recursos), las prioridades nodales, estado de los nodos, recursos, procesos y del sistema para producir una solución global satisfactoria. Este método evalúa esta información, para determinar cuál es el escenario según la carga de trabajo proporcionada, y en base a ello elige el operador de agregación correspondiente (ver Fig. 17).

En el modelo de decisión propuesto se ejecuta el primer operador de agregación, que corresponde al escenario general (Función de Asignación en Sistemas Distribuidos, **FASD**). Ver capítulo 2, correspondiente al escenario 1.

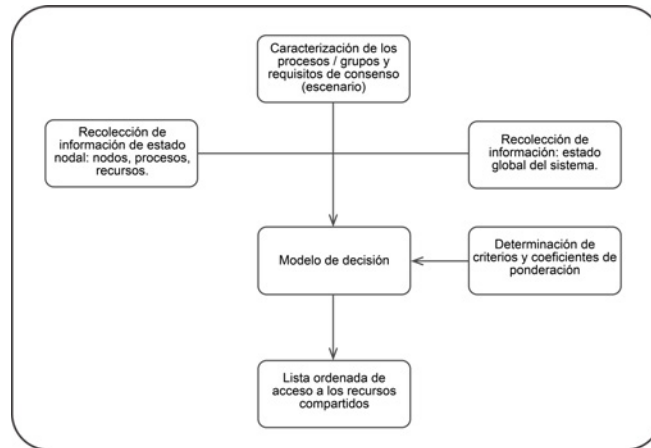


Fig. 17. Modelo global de decisión para acceso a recursos compartidos
Fuente: Elaboración propia.

De acuerdo a la información ingresada, se podrá seguir con el escenario general o hacer uso de uno de los otros operadores para los siguientes escenarios, la Función de Asignación en Sistemas Distribuidos Concatenada Ordenada, **FASDCO** (ver capítulo 3, correspondiente al escenario 2), la Función de Asignación para Sistemas Distribuidos Ordenada por Grupo Compatible, **FASDOGC** (ver capítulo 4, correspondiente al escenario 3), la Función de Asignación para Sistemas Distribuidos Ordenada por Grupo, **FASDOG** (ver capítulo 5, correspondiente al escenario 4). Seguidamente se deberá aplicar el proceso de resolución. Ver Fig. 18.

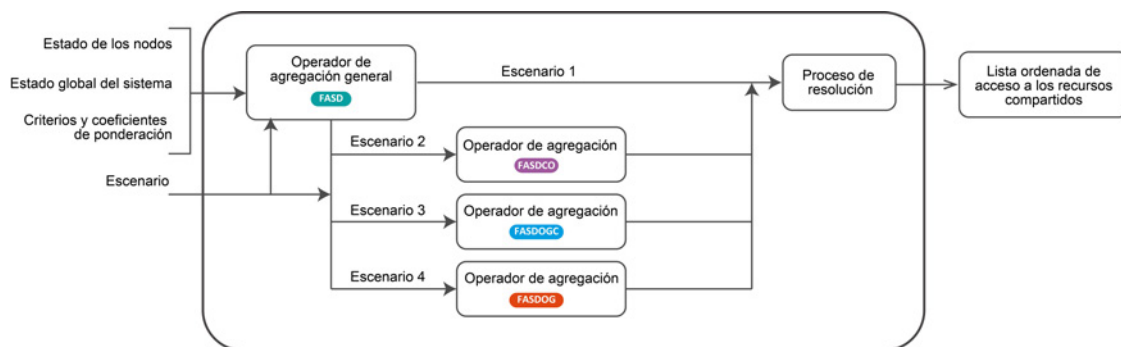


Fig. 18. Modelo de decisión propuesto
Fuente: Elaboración propia.

6.3. Consideraciones del modelo de decisión propuesto

En el modelo propuesto, 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 del 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 existe un *Runtime* coordinador global en uno de los nodos que evalúa y ejecuta el modelo de decisión y el operador de agregación correspondiente, ver Fig. 19.

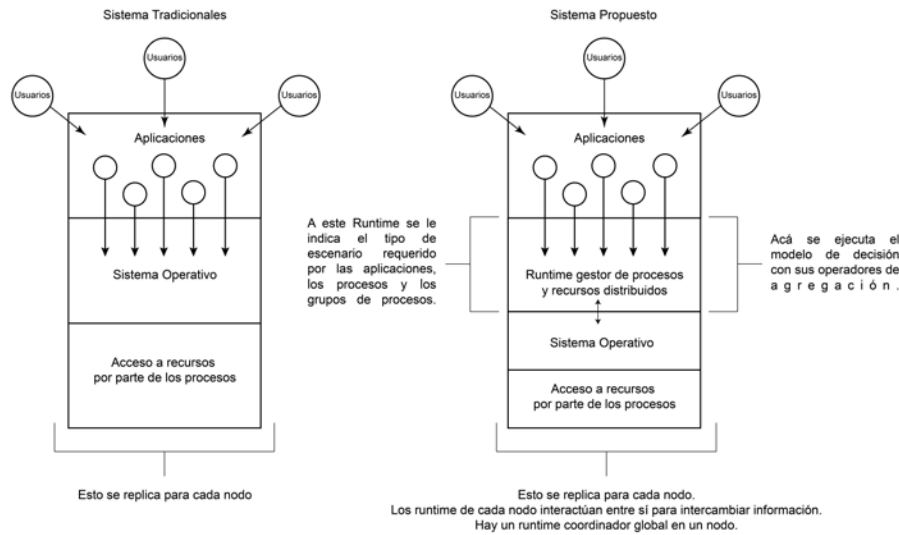


Fig. 19. Comparación del Modelo propuesto y los modelos tradicionales
Fuente: Elaboración propia.

6.1. Ejemplo de modelo de decisión aplicado a alguno de los algoritmos tradicionales

En este ejemplo se podrá visualizar cómo alguno de los métodos considerados tradicionales en este trabajo, son un caso particular del modelo de decisión desarrollado. Teniendo en cuenta que los métodos tradicionales no consideran los grupos de procesos, para los pesos asignados a los procesos en el cálculo de prioridades de la Tabla 109, habrá que inhabilitar la columna que considera si un proceso integra un grupo de procesos, es decir, que el cálculo sólo tendrá en cuenta que los procesos son independientes.

TABLA 109. Pesos asignados a los procesos para el cálculo de prioridades

Procesos	Pesos Finales	
	Si integra un grupo de procesos	Si es independiente
p_{11}	-	$wf_{11}=1/np$
....	-
p_{ki}	-	$wf_{ki}=1/np$
....	-	
p_{np}	-	$wf_{np}=1/np$

Nota: Pesos asignados a los procesos para calcular la prioridad o preferencia final de acceso a los recursos.

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Luego, respecto de los pesos asignados a los criterios de la Tabla 110, para calcular la prioridad, solo se tendrá en cuenta el criterio “Prioridad Proceso”, inhabilitando los demás criterios, dado que estos métodos no consideran ninguno de los criterios contemplados en el modelo propuesto.

Los vectores de valoraciones se aplican para cada requerimiento de un recurso hecho por un proceso, según los criterios establecidos para la determinación de la prioridad que en cada caso y momento fija el nodo en el cual se produce el requerimiento; cada vector de valoraciones de cada requerimiento se multiplica escalarmente por el vector de pesos correspondiente a la categoría de carga actual del nodo para obtener la prioridad según cada criterio y la prioridad nodal otorgada a cada requerimiento.

TABLA 110. Pesos asignados a los criterios para calcular la prioridad

Categorías	Pesos				Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S
	N° Proc.	% CPU	% Mem.	% MV				
Alta	-	-	-	-	0.1000	-	-	-
Media	-	-	-	-	0.2000	-	-	-
Baja	-	-	-	-	0.1000	-	-	-

Nota: Pesos asignados a los criterios para calcular la prioridad o preferencia que cada nodo otorgará a cada requerimiento de cada proceso según la carga del nodo.
 Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Si bien el modelo de decisión obtiene la información de todos los criterios, cabe destacar que para los métodos tradicionales, del vector de pesos (Tabla 110), sólo va a influir en la prioridad, el criterio “Prioridad Proceso”, esto se muestra en la Tabla 111.

TABLA 111. Valoraciones asignadas a los criterios para el cálculo de prioridad

Procesos - Recursos	Criterios				Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S	Prioridades Nodales
	N° Proc.	% CPU	% Mem.	% MV					
p_{11r11}	0.7000	0.5000	0.7000	0.9000	0.8000	0.2000	0.3000	0.4000	
$Pri(p_{11r11})$	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p_{11r12}	0.8000	0.7000	0.4000	0.5000	0.3000	0.7000	0.2000	0.4000	
$Pri(p_{11r12})$	0.0000	0.0000	0.0000	0.0000	0.0300	0.0000	0.0000	0.0000	0.0300
p_{11r21}	0.3000	0.4000	0.5000	0.2000	0.9000	0.2000	0.5000	0.7000	
$Pri(p_{11r21})$	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p_{11r22}	0.5000	0.5000	0.7000	0.4000	0.8000	0.3000	0.5000	0.6000	
$Pri(p_{11r22})$	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p_{11r23}	0.5000	0.6000	0.8000	0.8000	0.9500	0.9000	0.7000	0.6000	
$Pri(p_{11r23})$	0.0000	0.0000	0.0000	0.0000	0.0950	0.0000	0.0000	0.0000	0.0950
p_{11r24}	0.3000	0.5000	0.9000	0.2000	0.6000	0.6000	0.7000	0.4000	
$Pri(p_{11r24})$	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p_{12r11}	0.4000	0.7000	0.5000	0.9000	1.0000	0.9000	0.8000	0.8000	
$Pri(p_{12r11})$	0.0000	0.0000	0.0000	0.0000	0.1000	0.0000	0.0000	0.0000	0.1000
p_{12r12}	0.2000	0.7000	0.3000	0.7000	0.8000	0.3000	0.8000	0.9000	

TABLA 111. Valoraciones asignadas a los criterios para el cálculo de prioridad

Procesos - Recursos	Criterios								Prioridades Nodales
	N° Proc.	% CPU	% Mem.	% MV	Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S	
Pri(p12f12)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p12f21	0.7000	0.4000	0.3000	0.7000	0.8000	0.9000	0.5000	0.2000	
Pri(p12f21)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p12f22	0.9000	0.6000	0.7000	0.7000	0.8000	0.2000	0.5000	0.4000	
Pri(p12f22)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p12f31	0.2000	0.5000	0.7000	0.7000	0.3000	0.2000	0.7000	0.8000	
Pri(p12f31)	0.0000	0.0000	0.0000	0.0000	0.0300	0.0000	0.0000	0.0000	0.0300
p12f33	0.4000	0.5000	0.7000	0.9000	0.3000	0.4000	0.5000	0.8000	
Pri(p12f33)	0.0000	0.0000	0.0000	0.0000	0.0300	0.0000	0.0000	0.0000	0.0300
p13f11	0.5000	0.7000	0.7000	0.8000	0.6000	0.5000	0.7000	0.8000	
Pri(p13f11)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p13f12	0.7000	0.8000	0.7000	0.4000	0.9000	0.2000	0.9000	0.7000	
Pri(p13f12)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p13f13	0.7000	0.6000	0.7000	0.8000	0.9000	0.4000	0.8000	0.7000	
Pri(p13f13)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p13f21	0.7000	0.4000	0.9000	0.3000	0.5000	0.7000	0.2000	0.3000	
Pri(p13f21)	0.0000	0.0000	0.0000	0.0000	0.0500	0.0000	0.0000	0.0000	0.0500
p13f22	0.5000	0.9000	0.8000	0.3000	0.5000	0.4000	0.9000	0.3000	
Pri(p13f22)	0.0000	0.0000	0.0000	0.0000	0.0500	0.0000	0.0000	0.0000	0.0500
p13f31	0.5000	0.7000	0.3000	0.6000	0.8000	0.9000	0.9000	0.5000	
Pri(p13f31)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p13f32	0.6000	0.9000	0.3000	0.6000	0.4000	0.8000	0.7000	0.8000	
Pri(p13f32)	0.0000	0.0000	0.0000	0.0000	0.0400	0.0000	0.0000	0.0000	0.0400
p13f33	0.6000	0.2000	0.4000	0.6000	0.9000	0.8000	0.5000	0.8000	
Pri(p13f33)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p21f12	0.2000	0.1000	0.3000	0.8000	0.7000	0.7000	0.5000	0.8000	
Pri(p21f12)	0.0000	0.0000	0.0000	0.0000	0.1400	0.0000	0.0000	0.0000	0.1400
p21f13	0.2000	0.4000	0.8000	0.9000	0.7000	0.4000	0.5000	0.2000	
Pri(p21f13)	0.0000	0.0000	0.0000	0.0000	0.1400	0.0000	0.0000	0.0000	0.1400
p21f22	0.3000	0.5000	0.8000	0.9000	0.5000	0.6000	0.2000	0.4000	
Pri(p21f22)	0.0000	0.0000	0.0000	0.0000	0.1000	0.0000	0.0000	0.0000	0.1000
p21f23	0.7000	0.5000	0.6000	0.8000	0.5000	0.2000	0.9000	0.4000	
Pri(p21f23)	0.0000	0.0000	0.0000	0.0000	0.1000	0.0000	0.0000	0.0000	0.1000
p21f31	0.7000	0.5000	0.8000	0.6000	0.9000	0.4000	0.9000	0.9000	
Pri(p21f31)	0.0000	0.0000	0.0000	0.0000	0.1800	0.0000	0.0000	0.0000	0.1800
p21f33	0.7000	0.4000	0.9000	0.8000	0.4000	0.8000	0.6000	0.6000	
Pri(p21f33)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p22f21	0.5000	0.4000	0.7000	0.8000	0.6000	0.4000	0.6000	0.9000	
Pri(p22f21)	0.0000	0.0000	0.0000	0.0000	0.1200	0.0000	0.0000	0.0000	0.1200
p22f22	0.5000	0.9000	0.6000	0.8000	0.9000	0.4000	0.2000	0.1000	
Pri(p22f22)	0.0000	0.0000	0.0000	0.0000	0.1800	0.0000	0.0000	0.0000	0.1800
p22f31	0.5000	0.4000	0.5000	0.2000	0.4000	0.8000	0.2000	0.9000	
Pri(p22f31)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p22f33	0.8000	0.4000	0.3000	0.2000	0.8000	0.9000	0.5000	0.8000	
Pri(p22f33)	0.0000	0.0000	0.0000	0.0000	0.1600	0.0000	0.0000	0.0000	0.1600
p23f12	0.5000	0.6000	0.8000	0.3000	0.5000	0.7000	0.5000	0.5000	
Pri(p23f12)	0.0000	0.0000	0.0000	0.0000	0.1000	0.0000	0.0000	0.0000	0.1000
p23f24	0.6000	0.2000	0.1000	0.7000	0.3000	0.8000	0.9000	0.4000	
Pri(p23f24)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p23f31	0.3000	0.1000	0.4000	0.8000	0.7000	0.9000	0.4000	0.6000	
Pri(p23f31)	0.0000	0.0000	0.0000	0.0000	0.1400	0.0000	0.0000	0.0000	0.1400
p23f32	0.4000	0.4000	0.8000	0.2000	0.4000	0.6000	0.6000	0.6000	
Pri(p23f32)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p23f33	0.3000	0.6000	0.8000	0.2000	0.9000	0.6000	0.4000	0.2000	
Pri(p23f33)	0.0000	0.0000	0.0000	0.0000	0.1800	0.0000	0.0000	0.0000	0.1800
p24f11	0.4000	0.6000	0.7000	0.9000	0.5000	0.7000	0.9000	0.5000	
Pri(p24f11)	0.0000	0.0000	0.0000	0.0000	0.1000	0.0000	0.0000	0.0000	0.1000
p24f12	0.3000	0.6000	0.7000	0.8000	0.9000	0.7000	0.6000	0.9000	
Pri(p24f12)	0.0000	0.0000	0.0000	0.0000	0.1800	0.0000	0.0000	0.0000	0.1800
p24f23	0.4000	0.9000	0.4000	0.8000	0.8000	0.7000	0.6000	0.3000	
Pri(p24f23)	0.0000	0.0000	0.0000	0.0000	0.1600	0.0000	0.0000	0.0000	0.1600
p24f24	0.5000	0.8000	0.7000	0.9000	0.3000	0.4000	0.8000	0.7000	

TABLA 111. Valoraciones asignadas a los criterios para el cálculo de prioridad

Procesos - Recursos	Criterios								Prioridades Nodales
	N° Proc.	% CPU	% Mem.	% MV	Prioridad Proc.	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S	
Pri(p24f24)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p25f21	0.2000	0.8000	0.8000	0.9000	0.4000	0.5000	0.6000	0.8000	
Pri(p25f21)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p31f13	0.6000	0.9000	0.6000	0.9000	0.7000	0.4000	0.9000	0.8000	
Pri(p31f13)	0.0000	0.0000	0.0000	0.0000	0.0700	0.0000	0.0000	0.0000	0.0700
p31f31	0.8000	0.3000	0.9000	0.5000	0.7000	0.3000	0.9000	0.7000	
Pri(p31f31)	0.0000	0.0000	0.0000	0.0000	0.0700	0.0000	0.0000	0.0000	0.0700
p31f33	0.4000	0.7000	0.7000	0.5000	0.9000	0.9000	0.7000	0.7000	
Pri(p31f33)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p32f11	0.6000	0.9000	0.8000	0.5000	0.9000	0.7000	0.3000	0.7000	
Pri(p32f11)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p32f12	0.7000	0.4000	0.6000	0.9000	0.8000	1.0000	0.9000	0.7000	
Pri(p32f12)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p32f13	0.8000	0.9000	1.0000	0.7000	0.9000	0.3000	0.5000	0.9000	
Pri(p32f13)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p32f23	0.8000	0.8000	1.0000	0.9000	0.6000	0.8000	0.4000	0.9000	
Pri(p32f23)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p33f11	0.2000	0.7000	0.9000	0.8000	0.6000	0.9000	1.0000	0.3000	
Pri(p33f11)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p33f12	0.9000	1.0000	0.9000	0.7000	0.3000	0.5000	0.7000	0.3000	
Pri(p33f12)	0.0000	0.0000	0.0000	0.0000	0.0300	0.0000	0.0000	0.0000	0.0300
p33f13	0.9000	0.5000	0.7000	0.9000	0.3000	0.4000	0.5000	0.8000	
Pri(p33f13)	0.0000	0.0000	0.0000	0.0000	0.0300	0.0000	0.0000	0.0000	0.0300
p33f21	0.4000	0.6000	0.7000	0.8000	0.8000	0.4000	0.8000	0.8000	
Pri(p33f21)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p33f22	0.9000	0.4000	0.7000	0.8000	0.7000	0.4000	0.9000	0.7000	
Pri(p33f22)	0.0000	0.0000	0.0000	0.0000	0.0700	0.0000	0.0000	0.0000	0.0700
p33f23	0.4000	0.6000	0.9000	1.0000	0.6000	0.4000	0.7000	0.8000	
Pri(p33f23)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p33f32	0.6000	0.6000	0.7000	0.8000	0.4000	0.5000	0.8000	0.8000	
Pri(p33f32)	0.0000	0.0000	0.0000	0.0000	0.0400	0.0000	0.0000	0.0000	0.0400
p33f33	0.6000	1.0000	0.7000	0.4000	0.6000	0.8000	0.8000	0.9000	
Pri(p33f33)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p34f12	0.8000	1.0000	0.3000	0.5000	0.7000	0.3000	0.8000	0.7000	
Pri(p34f12)	0.0000	0.0000	0.0000	0.0000	0.0700	0.0000	0.0000	0.0000	0.0700
p34f13	0.2000	1.0000	0.8000	0.5000	0.8000	0.6000	0.9000	0.7000	
Pri(p34f13)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p34f22	0.9000	0.8000	0.6000	0.8000	0.9000	0.8000	0.5000	0.6000	
Pri(p34f22)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p34f23	0.4000	0.6000	0.7000	0.8000	0.9000	0.5000	0.9000	0.6000	
Pri(p34f23)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p34f24	0.9000	0.4000	0.7000	0.4000	0.7000	0.5000	0.9000	0.7000	
Pri(p34f24)	0.0000	0.0000	0.0000	0.0000	0.0700	0.0000	0.0000	0.0000	0.0700
p34f31	0.5000	0.6000	0.7000	0.9000	0.7000	0.9000	0.6000	0.7000	
Pri(p34f31)	0.0000	0.0000	0.0000	0.0000	0.0700	0.0000	0.0000	0.0000	0.0700
p34f32	0.8000	0.6000	0.9000	0.5000	0.6000	0.9000	0.3000	0.9000	
Pri(p34f32)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p34f33	0.4000	0.6000	0.8000	0.5000	0.9000	0.9000	0.4000	0.3000	
Pri(p34f33)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p35f12	0.2000	0.4000	0.7000	0.4000	0.9000	0.5000	0.7000	0.9000	
Pri(p35f12)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p35f13	0.8000	0.9000	0.7000	0.3000	0.9000	0.8000	0.7000	0.4000	
Pri(p35f13)	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p35f22	0.3000	0.8000	0.9000	0.4000	0.8000	0.6000	0.3000	0.6000	
Pri(p35f22)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p35f24	0.5000	0.8000	0.9000	0.4000	0.6000	0.9000	0.4000	0.7000	
Pri(p35f24)	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p35f31	0.9000	0.8000	0.7000	0.4000	0.8000	0.3000	0.4000	0.8000	
Pri(p35f31)	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p35f32	0.9000	0.7000	0.8000	0.6000	0.4000	0.9000	0.4000	0.7000	
Pri(p35f32)	0.0000	0.0000	0.0000	0.0000	0.0400	0.0000	0.0000	0.0000	0.0400
p35f33	0.5000	0.7000	0.6000	0.9000	0.8000	0.5000	0.9000	0.7000	

TABLA 111. Valoraciones asignadas a los criterios para el cálculo de prioridad

Procesos	Criterios								Prioridades Nodales
	N°	%	%	% MV	Prioridad	Sobrec.	Sobrec.	Sobrec.	
- Recursos	Proc.	CPU	Mem.		Proc.	Mem.	Proc.	E/S	
Pri(p _{35f33})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{36f11}	0.8000	0.9000	0.6000	0.5000	0.8000	0.9000	0.9000	0.6000	
Pri(p _{36f11})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{36f12}	0.7000	0.9000	0.4000	0.9000	0.8000	0.9000	0.4000	0.7000	
Pri(p _{36f12})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{36f13}	0.9000	0.9000	0.8000	0.7000	0.8000	0.7000	0.9000	0.7000	
Pri(p _{36f13})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{36f21}	0.8000	0.9000	0.5000	0.3000	0.7000	0.7000	0.9000	0.9000	
Pri(p _{36f21})	0.0000	0.0000	0.0000	0.0000	0.0700	0.0000	0.0000	0.0000	0.0700
p _{36f22}	0.8000	0.2000	0.1000	0.3000	0.8000	0.7000	0.6000	0.9000	
Pri(p _{36f22})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{36f24}	0.4000	0.7000	0.9000	0.3000	0.8000	0.5000	0.8000	0.9000	
Pri(p _{36f24})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{36f31}	0.5000	0.9000	0.9000	0.7000	0.8000	0.8000	0.8000	0.7000	
Pri(p _{36f31})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{36f32}	0.9000	0.8000	0.6000	0.7000	0.8000	0.5000	0.6000	0.7000	
Pri(p _{36f32})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{36f33}	0.2000	0.7000	0.4000	0.8000	0.8000	0.9000	0.4000	0.5000	
Pri(p _{36f33})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{37f11}	0.9000	0.6000	0.8000	0.6000	0.9000	0.7000	0.9000	0.8000	
Pri(p _{37f11})	0.0000	0.0000	0.0000	0.0000	0.0900	0.0000	0.0000	0.0000	0.0900
p _{37f12}	0.7000	0.9000	0.8000	0.7000	0.5000	0.8000	0.8000	0.6000	
Pri(p _{37f12})	0.0000	0.0000	0.0000	0.0000	0.0500	0.0000	0.0000	0.0000	0.0500
p _{37f21}	0.9000	0.7000	0.8000	0.6000	0.6000	0.9000	0.5000	0.4000	
Pri(p _{37f21})	0.0000	0.0000	0.0000	0.0000	0.0600	0.0000	0.0000	0.0000	0.0600
p _{37f32}	0.8000	0.9000	0.5000	0.3000	0.8000	0.7000	0.4000	0.6000	
Pri(p _{37f32})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800
p _{37f33}	0.8000	0.4000	0.6000	0.8000	0.8000	0.6000	0.9000	0.3000	
Pri(p _{37f33})	0.0000	0.0000	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0800

Nota: Valoraciones asignadas a los criterios para calcular la prioridad o preferencia nodal que cada nodo otorgará a cada requerimiento de cada proceso según la carga del nodo.
Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Se debe calcular las prioridades nodales, pesos finales y prioridades globales de los procesos para acceder a los recursos. La Tabla 112 se construye a partir de los valores de prioridades nodales de la Tabla 111. Como se había mencionado, los métodos tradicionales no consideran los grupos de procesos, el cálculo sólo tendrá en cuenta que los procesos son independientes. Dado que en el ejemplo existen 15 procesos, y que el cálculo de los pesos es wf_{ij} igual a $1/np$ para procesos independientes, donde np es el número de procesos en el sistema (15), el cálculo para los pesos de cada proceso (wp_{ij}) es igual a $1/15$. Para el cálculo de los pesos normalizados (nwp_{ij}) se divide cada valor de wp_{ij} por la sumatoria de todos los wp_{ij} , esto se puede observar en la Tabla 113.

TABLA 112. Prioridades nodales de los procesos para acceder a cada recurso

Recursos	Prioridades Nodales de los Procesos																
	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}		
r_{11}	0.0800	0.1000	0.0600	0.0000	0.0000	0.0000	0.1000	0.0000	0.0000	0.0900	0.0600	0.0000	0.0000	0.0800	0.0900		
r_{12}	0.0300	0.0800	0.0900	0.1400	0.0000	0.1000	0.1800	0.0000	0.0000	0.0800	0.0300	0.0700	0.0900	0.0800	0.0500		
r_{13}	0.0000	0.0000	0.0900	0.1400	0.0000	0.0000	0.0000	0.0000	0.0700	0.0900	0.0300	0.0800	0.0900	0.0800	0.0000		
r_{21}	0.0900	0.0800	0.0500	0.0000	0.1200	0.0000	0.0000	0.0800	0.0000	0.0000	0.0800	0.0000	0.0000	0.0700	0.0600		
r_{22}	0.0800	0.0800	0.0500	0.1000	0.1800	0.0000	0.0000	0.0000	0.0000	0.0000	0.0700	0.0900	0.0800	0.0800	0.0000		
r_{23}	0.0950	0.0000	0.0000	0.1000	0.0000	0.0000	0.1600	0.0000	0.0000	0.0600	0.0600	0.0900	0.0000	0.0000	0.0000		
r_{24}	0.0600	0.0000	0.0000	0.0000	0.0000	0.0600	0.0600	0.0000	0.0000	0.0000	0.0000	0.0700	0.0600	0.0800	0.0000		
r_{31}	0.0000	0.0300	0.0800	0.1800	0.0800	0.1400	0.0000	0.0000	0.0700	0.0000	0.0000	0.0700	0.0800	0.0800	0.0000		
r_{32}	0.0000	0.0000	0.0400	0.0000	0.0000	0.0800	0.0000	0.0000	0.0000	0.0000	0.0400	0.0600	0.0400	0.0800	0.0800		
r_{33}	0.0000	0.0300	0.0900	0.0800	0.1600	0.1800	0.0000	0.0000	0.0900	0.0000	0.0600	0.0900	0.0800	0.0800	0.0800		

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 113. Pesos finales y pesos finales normalizados

Recursos	Procesos																
	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}		
Pesos Finales	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666		
Pesos Finales Normalizados	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666	0.0666		

Nota: Pesos finales y pesos finales normalizados asignados a los procesos para calcular la prioridad o preferencia final de acceso a los recursos.
Fuente: Elaboración propia. Basado en La Red Martínez (2017).

TABLA 114. Prioridades globales finales de los procesos para acceder a cada recurso

Recursos	Prioridades Nodales de los Procesos																
	p_{11}	p_{12}	p_{13}	p_{21}	p_{22}	p_{23}	p_{24}	p_{25}	p_{31}	p_{32}	p_{33}	p_{34}	p_{35}	p_{36}	p_{37}		
r_{11}	0.0053	0.0067	0.0040	0.0000	0.0000	0.0000	0.0067	0.0000	0.0000	0.0060	0.0040	0.0000	0.0000	0.0053	0.0060		
r_{12}	0.0020	0.0053	0.0060	0.0093	0.0000	0.0067	0.0120	0.0000	0.0000	0.0053	0.0020	0.0047	0.0060	0.0053	0.0033		
r_{13}	0.0000	0.0000	0.0060	0.0093	0.0000	0.0000	0.0000	0.0000	0.0047	0.0060	0.0020	0.0053	0.0060	0.0053	0.0000		
r_{21}	0.0060	0.0053	0.0033	0.0000	0.0080	0.0000	0.0000	0.0053	0.0000	0.0000	0.0053	0.0000	0.0000	0.0047	0.0040		
r_{22}	0.0053	0.0053	0.0033	0.0067	0.0120	0.0000	0.0000	0.0000	0.0000	0.0000	0.0047	0.0060	0.0053	0.0053	0.0000		
r_{23}	0.0063	0.0000	0.0000	0.0067	0.0000	0.0000	0.0107	0.0000	0.0000	0.0040	0.0040	0.0060	0.0000	0.0000	0.0000		
r_{24}	0.0040	0.0000	0.0000	0.0000	0.0000	0.0040	0.0040	0.0000	0.0000	0.0000	0.0000	0.0047	0.0040	0.0053	0.0000		
r_{31}	0.0000	0.0020	0.0053	0.0120	0.0053	0.0093	0.0000	0.0000	0.0047	0.0000	0.0000	0.0047	0.0053	0.0053	0.0000		
r_{32}	0.0000	0.0000	0.0027	0.0000	0.0000	0.0053	0.0000	0.0000	0.0000	0.0000	0.0027	0.0040	0.0027	0.0053	0.0053		
r_{33}	0.0000	0.0020	0.0060	0.0053	0.0107	0.0120	0.0000	0.0000	0.0060	0.0000	0.0040	0.0060	0.0053	0.0053	0.0053		

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Para calcular las prioridades globales finales de los procesos para acceder a cada recurso, habrá que multiplicar cada valor de prioridad nodal de la Tabla 112 por su correspondiente en el vector de pesos normalizados (nwp_{ij}) de la Tabla 113. Esto se puede observar en la Tabla 114.

La sumatoria de todos los valores de asignación de recursos a procesos, de cada recurso, indicará la prioridad que tiene ese recurso de ser asignado, ver Tabla 115.

TABLA 115. Cálculo de prioridades de asignación de recursos

Recursos	Prioridad de asignación
$FSDPr_{11}$	0.0440
$FSDPr_{12}$	0.0680
$FSDPr_{13}$	0.0446
$FSDPr_{21}$	0.0420
$FSDPr_{22}$	0.0540
$FSDPr_{23}$	0.0376
$FSDPr_{24}$	0.0260
$FSDPr_{31}$	0.0540
$FSDPr_{32}$	0.0280
$FSDPr_{33}$	0.0680

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Ordenando la Tabla 115 de mayor a menor, se obtendrá el orden de asignación de recursos. Esto se puede observar en la Tabla 116.

TABLA 116. Cálculo de prioridades de asignación de recursos ordenada

Recursos	Prioridad de asignación
$FSDPr_{33}$	0.0680
$FSDPr_{12}$	0.0680
$FSDPr_{22}$	0.0540
$FSDPr_{31}$	0.0540
$FSDPr_{13}$	0.0446
$FSDPr_{11}$	0.0440
$FSDPr_{21}$	0.0420
$FSDPr_{23}$	0.0376
$FSDPr_{32}$	0.0280
$FSDPr_{24}$	0.0260

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

La Tabla 114 indicará cuál será el proceso con mayor prioridad para acceder a cada recurso, y la Tabla 116 el orden de los recursos a asignar. La Tabla 117 establecerá el orden final de asignación de recursos a procesos (**FASD**). Algunas consideraciones de la Tabla 117. Para la asignación del recurso r_{11} hay dos procesos p_{12} y p_{24} que tienen la misma prioridad calculada en la Tabla 114. Dado que los métodos tradicionales sólo tienen en cuenta la prioridad inicial del proceso, habrá que recurrir a

la Tabla 111 y observar el criterio “Prioridad Proceso” que indica la prioridad inicial del proceso, en la cual la asignación $p_{12}r_{11}$ tiene prioridad 1, mientras que la asignación $p_{24}r_{11}$ tiene prioridad 0.5.

TABLA 117. Orden o prioridad final de asignación (FASD)

Prioridad Global Final Ordenada	Asignación
0.0680	r_{33} al p_{23}
0.0680	r_{12} al p_{24}
0.0540	r_{22} al p_{22}
0.0540	r_{31} al p_{21}
0.0446	r_{13} al p_{21}
0.0440	r_{11} al p_{12}
0.0420	r_{21} al p_{22}
0.0376	r_{23} al p_{24}
0.0280	r_{32} al p_{36}
0.0260	r_{24} al p_{36}

Nota: Orden o prioridad final de asignación de los recursos y proceso al cual se asigna cada recurso **FASDO**.
Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Para la asignación del recurso r_{32} hay tres procesos p_{23} , p_{36} y p_{37} que tienen la misma prioridad calculada, por lo tanto, habrá que recurrir a la Tabla 111 y observar el criterio “Prioridad Proceso” que indica la prioridad inicial del proceso, en la cual la asignación $p_{23}r_{32}$ tiene prioridad 0.4 mientras que las asignaciones $p_{36}r_{32}$ y $p_{37}r_{32}$ tienen prioridad 0.8. Se considera que el p_{36} solicitó primero el recurso por lo que le es asignado.

Comparación de los resultados obtenidos con los métodos tradicionales y el método propuesto

Todas las tablas mencionadas a continuación, pertenecen al ejemplo del primer escenario (E1). Se compara cada elemento de la Tabla 117, con su correspondiente en las tablas del escenario E1.

El primer elemento de la Tabla 117, asignación del r_{33} al p_{23} , es la única que ocurre en la misma iteración (primera), aunque en el método desarrollado se coloca en la segunda posición, ver Tabla 118.

TABLA 118. Orden o prioridad final de asignación

Prioridad Global Final Ordenada	Asignación
0.47306452	r_{12} al p_{37}
0.46798387	r_{33} al p_{23}
0.37048387	r_{31} al p_{34}
0.35120968	r_{11} al p_{37}

0.34403226	r_{22} al p_{34}
0.33000000	r_{21} al p_{37}
0.32862903	r_{13} al p_{13}
0.30322581	r_{32} al p_{34}
0.24919355	r_{23} al p_{11}
0.18951613	r_{24} al p_{34}

Nota: Orden o prioridad final de asignación de los recursos y proceso al cual se asigna cada recurso **FASDO**.

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

La asignación del recurso r_{12} al proceso p_{24} se hace en la décima iteración, ver Tabla 119.

TABLA 119. Orden o prioridad final de asignación (décima iteración)

Prioridad Global Final Ordenada	Asignación
0.06032258	r_{12} al p_{24}
0.04250000	r_{33} al p_{35}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

En la Tabla 120 la asignación del recurso r_{22} al proceso p_{22} se hace en la sexta iteración.

TABLA 120. Orden o prioridad final de asignación (sexta iteración)

Prioridad Global Final Ordenada	Asignación
0.18282258	r_{33} al p_{31}
0.17669355	r_{12} al p_{12}
0.11411290	r_{31} al p_{12}
0.09943548	r_{22} al p_{22}
0.07741935	r_{21} al p_{11}
0.06983871	r_{13} al p_{36}
0.06604839	r_{11} al p_{36}
0.04322581	r_{32} al p_{35}
0.02056452	r_{23} al p_{24}
0.02024194	r_{24} al p_{24}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

En la Tabla 121 la asignación del recurso r_{31} al proceso p_{21} se hace en la tercera iteración.

TABLA 121. Orden o prioridad final de asignación (tercera iteración)

Prioridad Global Final Ordenada	Asignación
0.34677419	r_{33} al p_{13}
0.33443548	r_{12} al p_{23}
0.24250000	r_{31} al p_{21}
0.22330645	r_{22} al p_{13}
0.21233871	r_{11} al p_{13}
0.19983871	r_{21} al p_{13}
0.18612903	r_{13} al p_{31}
0.17524194	r_{32} al p_{13}
0.10790323	r_{23} al p_{21}
0.09516129	r_{24} al p_{23}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

En la Tabla 122 la asignación del recurso r_{13} al proceso p_{21} se hace en la cuarta

iteración.

TABLA 122. Orden o prioridad final de asignación (cuarta iteración)

Prioridad Global Final Ordenada	Asignación
0.28725806	r_{33} al p_{37}
0.27637097	r_{12} al p_{13}
0.19637097	r_{31} al p_{23}
0.17975806	r_{22} al p_{12}
0.15725806	r_{21} al p_{12}
0.14314516	r_{11} al p_{12}
0.13629032	r_{13} al p_{21}
0.11717742	r_{32} al p_{23}
0.07096774	r_{23} al p_{32}
0.06298387	r_{24} al p_{35}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

En la Tabla 123 la asignación del recurso r_{11} al proceso p_{12} se hace en la cuarta iteración.

TABLA 123. Orden o prioridad final de asignación (cuarta iteración)

Prioridad Global Final Ordenada	Asignación
0.28725806	r_{33} al p_{37}
0.27637097	r_{12} al p_{13}
0.19637097	r_{31} al p_{23}
0.17975806	r_{22} al p_{12}
0.15725806	r_{21} al p_{12}
0.14314516	r_{11} al p_{12}
0.13629032	r_{13} al p_{21}
0.11717742	r_{32} al p_{23}
0.07096774	r_{23} al p_{32}
0.06298387	r_{24} al p_{35}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

En la Tabla 124 la asignación del recurso r_{21} al proceso p_{22} se hace en la quinta iteración.

TABLA 124. Orden o prioridad final de asignación (quinta iteración)

Prioridad Global Final Ordenada	Asignación
0.22798387	r_{33} al p_{12}
0.22459677	r_{12} al p_{11}
0.15185484	r_{31} al p_{31}
0.13846774	r_{22} al p_{21}
0.11596774	r_{21} al p_{22}
0.09709677	r_{13} al p_{32}
0.08991935	r_{11} al p_{32}
0.06685484	r_{32} al p_{36}
0.04403226	r_{23} al p_{33}
0.04112903	r_{24} al p_{36}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

La Tabla 125 la asignación del recurso r_{23} al proceso p_{24} se hace en la sexta iteración.

TABLA 125. Orden o prioridad final de asignación (sexta iteración)

Prioridad Global Final Ordenada	Asignación
0.18282258	r_{33} al p_{31}
0.17669355	r_{12} al p_{12}
0.11411290	r_{31} al p_{12}
0.09943548	r_{22} al p_{22}
0.07741935	r_{21} al p_{11}
0.06983871	r_{13} al p_{36}
0.06604839	r_{11} al p_{36}
0.04322581	r_{32} al p_{35}
0.02056452	r_{23} al p_{24}
0.02024194	r_{24} al p_{24}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

En la Tabla 126 la asignación del recurso r_{32} al proceso p_{36} se hace en la quinta iteración.

TABLA 126. Orden o prioridad final de asignación (quinta iteración)

Prioridad Global Final Ordenada	Asignación
0.22798387	r_{33} al p_{12}
0.22459677	r_{12} al p_{11}
0.15185484	r_{31} al p_{31}
0.13846774	r_{22} al p_{21}
0.11596774	r_{21} al p_{22}
0.09709677	r_{13} al p_{32}
0.08991935	r_{11} al p_{32}
0.06685484	r_{32} al p_{36}
0.04403226	r_{23} al p_{33}
0.04112903	r_{24} al p_{36}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

La Tabla 127 la asignación del recurso r_{24} al proceso p_{36} se hace en la quinta iteración.

TABLA 127. Orden o prioridad final de asignación (quinta iteración)

Prioridad Global Final Ordenada	Asignación
0.22798387	r_{33} al p_{12}
0.22459677	r_{12} al p_{11}
0.15185484	r_{31} al p_{31}
0.13846774	r_{22} al p_{21}
0.11596774	r_{21} al p_{22}
0.09709677	r_{13} al p_{32}
0.08991935	r_{11} al p_{32}
0.06685484	r_{32} al p_{36}
0.04403226	r_{23} al p_{33}
0.04112903	r_{24} al p_{36}

Fuente: Elaboración propia. Basado en La Red Martínez (2017).

Se ha visto que en esta comparación, los resultados de asignaciones en los métodos tradicionales no son iguales que los del modelo propuesto, y esto se debe a que los métodos tradicionales contemplan un solo tipo de criterio (prioridad de proceso), y no tienen en cuenta el número de procesos, %CPU, %Mem, %MV, etc., es decir, la

carga de cada nodo y el estado global del sistema, ver Tabla 128.

TABLA 128. Cuadro comparativo del modelo propuesto y los tradicionales

Características	Modelo propuesto	Algoritmo centralizado	Algoritmo Distribuido de Lamport, Ricart y Agrawala	Algoritmo de anillo de fichas
Orden inicial	por prioridades	marca de tiempo	marca de tiempo	marca de tiempo
Solicitudes	cálculo de prioridad	mensajes al coordinador	mensajes a $n-1$ procesos	mensajes al vecino
Método de asignación	mayor prioridad calculada	menor marca de tiempo o prioridad inicial	menor marca de tiempo o prioridad inicial	menor marca de tiempo o prioridad inicial
Grupos de procesos	agrupación por grupos	procesos independientes	procesos independientes	procesos independientes
Predisposición (prioridad nodal), carga del nodo.	☑	×	×	×
Estado nodal (nodos, procesos, grupos, recursos.)	☑	×	×	×
Estado global del sistema.	☑	×	×	×

Fuente: Elaboración propia.

En este sentido, se puede decir que además de ser los métodos tradicionales, un caso particular del método propuesto, este nuevo modelo permite una evaluación más aproximada al estado real del sistema, lo que permitiría obtener mejores resultados en las asignaciones.

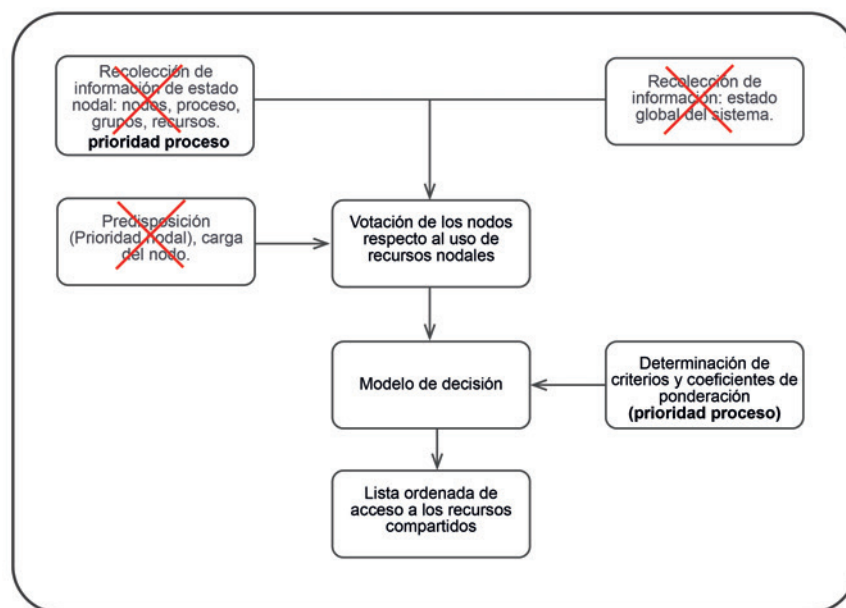


Fig. 20. Modelo global de decisión para accesos a recursos compartidos.

En la Fig. 20 se puede observar que en el modelo global, para el ejemplo de los métodos tradicionales, no se tiene en cuenta la recolección de información del estado global del sistema, ni la predisposición (prioridad nodal), ni la carga del nodo, sólo se contempla la prioridad de proceso.

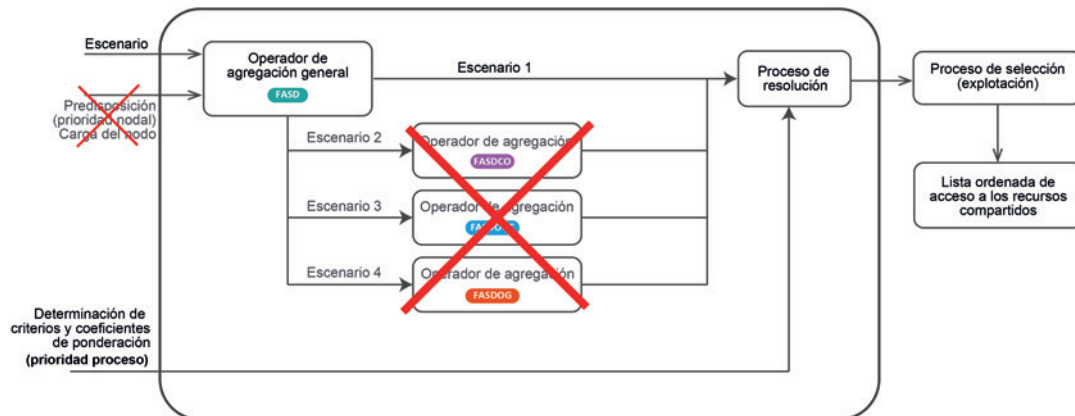


Fig. 21. Modelo de decisión incluyendo en el proceso de resolución los métodos de agregación específicos para al acceso a recursos compartidos.

Para la Fig. 21 el modelo de decisión, en el caso del ejemplo de los métodos tradicionales, no se tiene en cuenta la predisposición (prioridad nodal), ni la carga del nodo. La determinación de los criterios y coeficientes de ponderación se basan sólo en la prioridad de proceso. Además, no se contemplan los demás escenarios.

Cabe destacar que los resultados finales obtenidos se ajustan a cada escenario en particular. Es decir, al cambiar las condiciones del escenario, los resultados obtenidos en la aplicación del Modelo de Decisión pueden ser diferentes.

Algunas de las diferencias más significativas se pueden observar en la Tabla 128.

6.2. Consideraciones acerca de las operaciones de agregación

En (La Red Martínez, 2017) y (La Red Martínez et al. 2018) las características de las operaciones de agregación descritas permiten considerar que el método propuesto pertenece a la familia de operadores de agregación *Neat-OWA*, operadores que se caracterizan por lo siguiente:

La definición de los operadores *OWA* indica que $f(a_1, a_2, \dots, a_n) = \sum_{j=1}^n w_j \cdot b_j$,

donde b_j es el j -ésimo valor mayor de las a_n , con la restricción para los pesos de satisfacer (1) $w_i \in [0,1]$ y (2) $\sum_{i=1}^n w_i = 1$.

Para la familia de operadores *Neat OWA* los pesos serán calculados en función de los elementos que se agregan, o más exactamente de los valores a agregar ordenados, los b_j , manteniéndose las condiciones (1) y (2). Es este caso los pesos son: $w_i = f_i(b_1, \dots, b_n)$, definiéndose el operador

$$F(a_1, \dots, a_n) = \sum_i f_i(b_1, \dots, b_n) \cdot b_i$$

Para esta familia, donde los pesos dependen de la agregación, no se exige la satisfacción de todas las propiedades de los operadores *OWA*.

Además, para poder afirmar que un operador de agregación es *neat*, es necesario que el valor final de agregación sea independiente del orden de los valores. Sea $A = (a_1, \dots, a_n)$ las estradas a agregar, sea $B = (b_1, \dots, b_n)$ las entradas ordenadas y $C = (c_1, \dots, c_n) = Perm(a_1, \dots, a_n)$ una permutación de las entradas. Formalmente se define un operador *OWA* como *neat* si

$$F(a_1, a_2, \dots, a_n) = \sum_{i=1}^n w_i \cdot b_i$$

Produce el mismo resultado para cualquier asignación $C = B$.

6.3. Discusiones y comentarios

Se han definido las características del Modelo de Decisión para permitir definir los objetivos claramente, para estudiar las posibles alternativas y evaluar sus consecuencias. La tarea consistió en seleccionar la mejor alternativa para lograr la mejor optimización. El objetivo principal del modelo propuesto fue considerar el entorno de ejecución distribuida de procesos, que podrían estar agrupados, con distintas exigencias de consenso respecto del acceso a recursos compartidos, para seleccionar el método

de agregación más adecuado a cada escenario posible, y así, generar la secuencia de asignación de los recursos a los procesos que los solicitan, respetando la exclusión mutua en el acceso a dichos recursos. Se ha explicado que el modelo de decisión utiliza un Runtime que gestiona los procesos y recursos compartidos y define el escenario correspondiente. Además, se ha realizado un cuadro comparativo del modelo propuesto con respecto a los modelos tradicionales y se ha comentado las consideraciones de los operadores de agregación desarrollados.

La estructura de datos mencionada anteriormente y el método de agregación utilizado no están totalmente cubiertos por los métodos tradicionales, por ejemplo, no contemplan la predisposición (prioridad nodal), carga del nodo, el estado nodal (nodos, procesos, grupos, recursos.) ni el estado global del sistema, para el cálculo de prioridades en las asignaciones de recursos a procesos.

Una de las características a señalar de los operadores *Neat OWA* es que los valores a agregar no necesitan ser ordenados para su proceso. Esto implica que la formulación de un operador *neat* puede ser definida usando directamente los argumentos en lugar de los elementos ordenados.

En el operador de agregación propuesto los pesos se calculan en función de valores del contexto del cual surgen los valores a agregar.

Capítulo VII - Conclusiones y futuras líneas de investigación

7.1. Conclusiones

Capítulo I

En el capítulo I se ha hecho la descripción de antecedentes, el marco teórico, se han planteado las hipótesis, se han contemplado situaciones en las que los procesos accedieron a recursos compartidos en la modalidad de exclusión mutua con o sin constituir grupos de procesos, que podían requerir o no sincronización y que podían tener o no exigencias estrictas de consenso para lograr el acceso.

Se han descrito las estructuras de datos, los elementos que intervienen dentro del sistema, los nodos, recursos, procesos, que tienen prioridades y cargas de trabajo y que en diferentes situaciones permitieran que el sistema se auto regula reiteradamente en función del estado local de los n nodos y del estado global del sistema.

Capítulo II

En el capítulo II se han definido las hipótesis y objetivos planteados para el escenario general en el que los procesos accedieran a recursos compartidos en la modalidad de exclusión mutua pudiendo constituir grupos de procesos, los procesos no requerían sincronización (estar activos en sus respectivos procesadores en un mismo lapso de tiempo) y procesos no tenían exigencias estrictas de consenso para lograr el acceso a los recursos.

Se ha explicado el operador de agregación para este escenario, que parte de (La Red Martínez, 2017), y se ha desarrollado un ejemplo con los datos provenientes del mismo, hasta obtener la tabla **FASD** final de asignaciones de recursos a procesos, la cual es utilizada como punto de partida para los siguientes capítulos.

Capítulo III

En el capítulo III se han definido las hipótesis y objetivos planteados para el escenario 2, 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) y (La Red Martínez et al. 2018). Se ha establecido un modelo de consenso que favoreció el acceso secuencial de los procesos a todos los recursos solicitados. Las premisas, las estructuras de datos y el operador mencionado en el capítulo anterior se utilizaron como punto de partida para crear un nuevo operador en el escenario descrito en este capítulo. Utilizó la tabla **FASDC**, que es la concatenación de las tablas **FASD** ordenadas 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.

Capítulo IV

En el capítulo IV se han definido las hipótesis y objetivos planteados para el escenario 3, se buscó que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua constituyendo 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 requirió un consenso para asignar de manera consecutiva los recursos solicitados por un grupo de procesos, es decir, que iniciada la secuencia de asignación de recursos al grupo, la misma no pudo ser interrumpida para asignar recursos a otro grupo). Se ha presentado una variante de un método innovador para la gestión de recursos compartidos en sistemas distribuidos, basado en (La Red Martínez, 2017) y (La Red

Martínez et al. 2018), donde se desarrolló un operador de agregación, para asignar recursos en sistemas distribuidos, pero en este caso, se ha establecido un modelo de consenso que favoreció el acceso secuencial de los procesos de cada grupo a todos los recursos solicitados. Las premisas, las estructuras de datos y el operador mencionado en el segundo capítulo se utilizan como punto de partida para crear un nuevo operador. Utilizó la tabla **FASDCO**, que es la concatenación de las tablas **FASD** ordenadas por procesos de todas las rondas del escenario general y consideró el promedio global de prioridades que cada grupo de procesos tiene sobre todos los recursos de todas sus asignaciones en las diferentes rondas, pero para la asignación global final del grupo, estableció subrondas compatibles de asignación, dado que en un mismo grupo pudo haber procesos que requieran un mismo recurso, por lo tanto, los procesos de mayor prioridad fueron asignados en una primer subronda y los que generaban incompatibilidad se asignaron en una siguiente subronda.

Capítulo V

En el capítulo V se han definido las hipótesis y objetivos planteados para el escenario 4, se buscó que los procesos accedieran a recursos compartidos en la modalidad de exclusión mutua constituyendo grupos de procesos (los procesos independientes son considerados grupos unitarios); los procesos no requirieron sincronización y sin exigencias estrictas de consenso para lograr el acceso (no se requería un consenso para asignar de manera consecutiva los recursos solicitados por un procesos, es decir, que iniciada la secuencia de asignación de recursos al grupo, 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) y (La Red Martínez et al. 2018). Se ha establecido un orden de asignación por grupo de procesos en cada instancia. Las premisas, las estructuras de datos y el operador mencionado en el segundo capítulo se utilizaron como punto de partida para crear un nuevo operador

en este escenario. Se utilizó la tabla **FASDO** hasta obtener el orden o prioridad de asignación de los recursos y el proceso al que se asigna cada recurso ordenada por grupo **FASDOG**. El modelo propuesto logró establecer un orden que permitió a los grupos de procesos acceder a todos sus recursos de manera secuencial y que éstos no pudieron ser removidos hasta que el grupo de procesos que los mantenía los liberó. El orden de asignación fue determinado por la prioridad promedio general de cada grupo en cada asignación.

Capítulo VI

Se han definido las características del Modelo de Decisión para permitir definir los objetivos claramente, para estudiar las posibles alternativas y evaluar sus consecuencias. Se ha explicado que el modelo de decisión utiliza un Runtime que gestiona los procesos y recursos compartidos y define el escenario correspondiente. Además, se ha realizado un cuadro comparativo del modelo propuesto con respecto a los modelos tradicionales y se ha comentado las consideraciones de los operadores de agregación desarrollados, basándose en las características de los operadores *Neat* OWA.

7.2. Futuras líneas de investigación

Se considera desarrollar modelos de decisión desde la óptica cognitiva para la toma de decisiones en grupos de procesos, contemplando los principios de la cibernética de segundo orden, en el contexto de sistemas complejos de auto-regulación, que trasciendan el enfoque tradicional de las ciencias de la computación considerando la posibilidad de imputación de datos faltantes, por ejemplo, como consecuencia de problemas en las comunicaciones entre los procesos, y fuzzyficación de variables para dar soporte a situaciones donde no es posible o conveniente expresar valores exactos. Se considera utilizar lógica difusa con conjuntos de etiquetas lingüísticas de al menos 5 etiquetas.

Simulador

Se prevé desarrollar un simulador en el que se consideren los distintos escenarios posibles para permitir al sistema predecir, comparar y optimizar el comportamiento de sus procesos simulados en un tiempo muy breve sin el coste ni el riesgo de llevarlos a cabo, haciendo posible la representación de los procesos, recursos y nodos en un modelo dinámico. Con la ayuda del correspondiente soporte informático, el modelo de simulación permitirá la capacidad de considerar complejas tareas interrelacionadas y proyectarlas mediante la realización de muchas combinaciones alternativas en cuestión de segundos. Además, la interacción de los recursos con los procesos se traducirá en un gran número de escenarios y de posibles resultados imposibles de abarcar y valorar sin la ayuda de un modelo de simulación computarizado.

Seguridad

Considerar aspectos vinculados a la seguridad en la ejecución de los procesos, en el acceso a los recursos y en la comunicación entre los nodos.

Bibliografía

• 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). The Science and Information (SAI) Organization, England, U.K. Vol. 9, no. 12. <http://dx.doi.org/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". 10th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2019), Orlando, USA.

• Agostini F., La Red Martínez D. L., (2019). "Allocation of shared resources". 14^a Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI'2019). ISBN N° 978-989-98434-9-3, pp. 1-6. Universidad de Coimbra, Coimbra, Portugal.

• Agostini F., Fornerón Martínez J. T., La Red Martínez D. L., (2019). "Nuevo operador de agregación para grupos de procesos". 16^a Conferencia Ibero Americana. Lisboa, Portugal. (Aceptado).

• Agrawal, D. y El Abbadi, A. (1991). An Efficient and Fault-Tolerant Solution of Distributed Mutual Exclusion. ACM Trans. on Computer Systems. Vol. 9, pp. 1-20. USA.

• Alagarsamy, K. (2003). Some Myths About Famous Mutual Exclusion Algorithms. ACM SIGACT News, Vol. 34, no. 3, pp. 94-103.

• Bagrodia, R. (1989). Process synchronization: design and performance evaluation of distributed algorithms, IEEE Transactions on Software Engineering. Vol. 15, Issue 9, pp. 1053 - 1065.

• Bateson, G. (1991). Pasos Hacia Una Ecología de la Mente. Planeta – Carlos

Lohlé. Argentina.

- Birman, K. (2005). *Reliable Distributed Systems: Technologies, Web Services and Applications*. New York: Springer-Verlag.

- Birman, K. P., Joseph, T. (1987). *Reliable Communication in the Presence of Failures*. *ACM Trans. on Computer Systems*. Vol. 5, pp. 47-76. USA.

- Birman, K. P., Schiper, A. y Stephenson, P. (1991). *Lightweight Causal and Atomic Group Multicast*. *ACM Trans. on Computer Systems*. Vol. 9, pp. 272-314. USA.

- Brix, V. H. (1970). *You are a Computer: Cybernetics in Everyday Life*.

- Cao, G. and Singhal, M. (2001). *A Delay-Optimal Quorum-Based Mutual Exclusion Algorithm for Distributed Systems*. *IEEE Transactions on Parallel And Distributed Systems*. Vol. 12, no. 12, pp. 1256-1268. USA.

- Castillo, E.; La Red, D.L. (2005). *Algoritmo Óptimo para Planificación de Búsqueda en Disco*. CISC I 2005 (4ta. Conferencia Iberoamericana en Sistemas, Cibernética e Informática). *Memorias*. Vol. I, pp. 325-330. Florida, USA.

- Chao, X., Kou, G., Peng, Y. (2016). *An optimization model integrating different preference formats*, 6th International Conference on Computers Communications and Control (ICCCC), pp. 228 – 231.

- Chen, C. T. (2001). *Applying linguistic decision-making method to deal with service quality evaluation problems*. *International Journal of Uncertainty, Fuzzyness and Knowledge-Based Systems*, Vol. 9 (Suppl.), pp. 103-114.

- Chiclana, F., Herrera, F., Herrera-Viedma, E. (2000). *The Ordered Weighted Geometric Operator: Properties And Application*. In: *Proc of 8th International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems*.

pp 985-991. España

- Chiclana, F., Herrera, F., Herrera-Viedma, E. (2001). Integrating Multiplicative Preference Relations In A Multipurpose Decision Making Model Based On Fuzzy Preference Relations. *Fuzzy Sets and Systems*. Vol. 112, pp. 277-291.
- Chiclana, F., Herrera-Viedma, E., Herrera, F., Alonso, S. (2004). Induced Ordered Weighted Geometric Operators and Their Use in the Aggregation of Multiplicative Preferences Relations. *International Journal of Intelligent Systems*. Vol. 19, pp. 233-255.
- Dong, Y., Zhang, H., Herrera-Viedma, E. (2016b). Consensus reaching model in the complex and dynamic MAGDM problem, *Knowledge-Based Systems*. Vol. 106, pp. 206-219, Elsevier.
- Doña, J. M.; Gil, A. M.; Peláez, J. I.; La Red Martínez, D. L. (2011). A System Based on the Concept of Linguistic Majority for the Companies Valuation. *Revista Econo Quantum*. Vol. 8, no. 2, pp. 121-142. México.
- Fodor, J. C., Roubens, M. (1994). *Fuzzy Preference Modelling and Multicriteria Decision Support*. Kluwer, Dordrecht.
- Fullér, R. (1996). *OWA Operators in Decision Making*. En Carlsson, C. ed. *Exploring the Limits of Support Systems*, TUCS General Publications, Turku Centre for Computer Science. No. 3, pp. 85-104.
- García-Melón, M., Ferrís-Oñate, J., Aznar-Bellver, J., Poveda-Bautista, R. (2006). *Farmland Appraisal: An Analytic Network Process (ANP) Approach*. MCDM. Greece.
- Gardner, H. (1996). *La Nueva Ciencia de la Mente*. Ediciones Paidós, 2ª reimpresión. España.
- Greco, S.; Matarazzo, B., Slowinski, R. (2002). *Rough sets methodology for*

sorting problems in presence of multiple attributes and criteria. *European Journal of Operational Research*. Vol.138, pp. 247-259.

- Joseph, T.A. y Birman, K.P. (1989). *Reliable Broadcast Protocols*, en *Distributed Systems*. Mullender, S. (Ed). ACM Press. USA.

- Joshi, R., Holzmann, G. J. (2007), *A Mini-Challenge: Build a Verifiable Filesystem*, *Formal Aspects of Computing*. Vol. 19.

- Kaashoek, M. K. (1992). *Group Communication In Distributed Computer Systems*, Centrale Huisdrukkerij Vrije Universiteit, Amsterdam.

- La Red Martínez, D.L. (2004). *Sistemas Operativos*. EUDENE. Argentina.

- La Red Martínez, D.L. (2010a). *Hacia una Concepción Integrada de la Mente*. Año 7 N° 1. *Revista Enl@ce: Revista Venezolana de Información, Tecnología y Conocimiento*. Pp. 123-136. Venezuela.

- La Red Martínez, D.L. (2010b). *¿Puede la Mente Ser Tratada Como un Artefacto?*. Año 6 N° 23. *Revista Cognición (Fundación Latinoamericana Para la Educación a Distancia - Instituto Latinoamericano de Investigación Educativa)*. Pp. 16-29. Argentina.

- La Red Martínez, D. L., Acosta, C. (2015). *Aggregation Operators Review - Mathematical Properties and Behavioral Measures*. *International Journal of Intelligent Systems and Applications (IJISA)*. Vol. 7, no. 10, pp. 63-76. Hong Kong.

- La Red Martínez, D. L., Acosta, J. C. (2014). *Perspectives of New Decision Making Models of Processes Synchronization in Distributed Systems*, *International Journal of Management and Information Technologies*, Vol. 09, no. 1, pp. 1504-1512, ISSN N° 2278-5612, U.S.A.

- La Red Martínez, D. L., Acosta, J. C. (2015). Review of Modeling Preferences for Decision Models, Volume 11 – N° 36, European Scientific Journal (ESJ), pp. 1-18, ISSN N° 1857-7881, Macedonia.

- La Red Martínez, D. L., Acosta, J., Gerzel, S. M., Rambo, A. R. (2014). Nuevo Enfoque Para la Sincronización de Procesos en Sistemas Distribuidos, III Jornadas de Investigación en Ingeniería del NEA y Países Limítrofes - III JIIM, ISBN N° 978-950-42-0157-1, Universidad Tecnológica Nacional, Resistencia, Argentina.

- La Red Martínez, D. L., Agostini, F. (2013). Lively Learning Framework of ISO/OSI Model and Data Communications, International Journal of Computer and Information Technologies. Vol 02, Issue 06, pp. 1087-1094, ISSN N° 2279-0764, U.S.A.

- La Red Martínez, D. L., Agostini, F. (2014). ISO/OSI model and data communication by animations, Revista WIT Transactions on Information and Communication Technologies, WIT Press, Vol. 58, pp. 963-970, ISBN N° 978-1-84564-975-3, Ashurst, Southampton, U.K.

- La Red Martínez, D. L., Pinto, N. (2015). Brief Review of Aggregation Operators. Wulfenia Journal. Vol. 22, no. 4, pp. 114-137. Austria.

- La Red, D. L.; Doña, J. M.; Peláez, J. I.; Fernández, E. B. (2011a). WKC-OWA, a New Neat-OWA Operator to Aggregate Information in Democratic Decision Problems. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, World Scientific Publishing Company, pp. 759-779. Francia.

- La Red, D. L.; Peláez, J. I. (2002). Optimization of Disk Seek Operations Using Neural Networks. 31 JAIIO (Argentine Conference on Computer Science and Operational Research) - AST 2002 (Argentine Symposium on Computing Technology). Proceedings, pp. 119-129. Argentina.

- La Red Martínez D. L. (2017). Aggregation Operator for Assignment of Resources in Distributed Systems. *International Journal of Advanced Computer Science and Applications (IJACSA)*. The Science and Information (SAI) Organization, Vol. 8, no. 10, pp. 406-419. ISSN N° 2156-5570. England, U.K.
- La Red Martínez D. L., Agostini F. (2017) “Modelo de Asignación de Recursos para la Enseñanza de los Procesos Distribuidos”. Primer Congreso Latinoamericano de Ingeniería – CLADI. Centro Provincial de Convenciones Paraná, Entre Ríos, Argentina.
- La Red Martínez D. L., Agostini F., Primorac C. (2018). “Nueva propuesta para la administración de recursos y procesos en sistemas distribuidos”. Workshop de Investigadores en Ciencias de la Computación, WICC. ISBN 978-987-3619-27-4. Corrientes, Argentina.
- La Red Martínez D. L., Acosta J. C., Agostini F. (2018). “Assignment of Resources in Distributed Systems”. 9th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2018), Orlando, USA.
- La Red Martínez D. L., Acosta J. C., Agostini F. (2019). “Assignment of Resources in Distributed Systems With Strict Consensus Requirements”. 9th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2019), Orlando, USA.
- Lamport, L. (1978). Time, Clocks, and the Ordering of Events in a Distributed System, *Communications of the ACM*, Vol. 21, no. 7, pp. 558-565.
- Lodha, S. and Kshemkalyani, A. (2000). A Fair Distributed Mutual Exclusion Algorithm. *IEEE Trans. Parallel and Distributed Systems*. Vol. 11, no. 6, pp. 537-549. USA.
- Longo, G. (1972). *Selected Topics in Information Theory*.
- Lu, J.; Zhang, G.; Ruan, D., Wu, F. (2007). *Multi-objective Group Decision Making: Methods, Software and Applications with Fuzzy Set Technology*. London: Imperial College

Press.

- Macedonia, M. R., Zyda, M. J., Pratt, D. R., Brutzman, D. P., Barham, P. T. (1995). Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments. Proc. of IEEE VRAIS, pp. 2-10.
- Marakas, G. (2002). Decision Support Systems. New Jersey, USA. Prentice Hall.
- Martínez, L.; Liu, J.; Ruan, D., Yang, J. B. (2007). Dealing with heterogeneous information in engineering evaluation processes. Information Sciences. Vol. 177, no. 7, pp. 1533-1542.
- Martínez, L., Liu, J., Yang, J. B. (2006). A fuzzy model for design evaluation based on multiplecriteria analysis in engineering systems. International Journal of Uncertainty, Fuzziness and KnowledgeBased Systems. Vol. 14, no. 3, pp. 317-336.
- Parsegian V. L. (1973). This Cybernetic World of Men, Machines, and Earth Systems, Anchor Books: Garden City, New York.
- Peláez, J.I., Doña, J.M. (2003a). LAMA: A Linguistic Aggregation of Majority Additive Operator. International Journal of Intelligent Systems.
- Peláez, J.I., Doña, J.M. (2003b). Majority Additive-Ordered Weighting Averaging: A New Neat Ordered Weighting Averaging Operators Based on the Majority Process. International Journal of Intelligent Systems. Vol. 18, no. 4, pp. 469-481.
- Peláez, J.I., Doña, J.M., Gómez-Ruiz, J.A. (2007). Analysis of OWA Operators in Decision Making for Modelling the Majority Concept. Applied Mathematics and Computation. Vol. 186, pp. 1263-1275.
- Peláez, J.I., Doña, J.M., La Red Martínez, D.L. (2003). Analysis of the Majority Process in Group Decision Making Process. 9th International Conference on Fuzzy Theory

and Technology. North Carolina. USA.

- Peláez, J.I., Doña, J.M., La Red Martínez, D.L. (2009). A Mix Model of Discounted Cash-Flow and OWA Operators for Strategic Valuation. V. 1 N° 2. Revista Interactive Multimedia and Artificial Intelligence - Special Issue On Business Intelligence And Semantic Web. Pp. 20-25. España.

- Peláez, J.I., Doña, J.M., La Red Martínez, D.L., Mesas, A. (2004). Majority Opinion in Group Decision Making Using the QMA-OWA Operator. Proceeding of ESTYLF. Pp. 449-454.

- Renesse, R., Birman, K., Vogels, W. (2003). Astrolabe: A robust and scalable technology for distributed systems monitoring, management, and data mining. ACM Transactions on Computer Systems, Vol. 21, no. 3.

- Ricart, G. y Agrawala, A.K. (1981). An Optimal Algorithm for Mutual Exclusion in Computer Networks. Commun. of the ACM. Vol. 24, pp. 9-17.

- Saaty, T. L. (1980). The Analytic Hierarchy Process. MacGraw Hill. USA.

- Samaja, J. (1993). Epistemología y Metodología. EUDEBA. Argentina.

- Samaja, J. (2004). Diseño, Proceso y Proyecto. Ed. JVE. Argentina.

- Shannon, C. E., Warren Weaver (1949): A Mathematical Model of Communication. Urbana, IL: University of Illinois Press.

- Silberschatz, A., Galvin, P.B. y Gagne, G. (2006). Fundamentos de Sistemas Operativos. 7ma. Edición. McGraw-Hill / Interamericana de España. S.A.U. España.

- Stallings, W. (2005). Sistemas Operativos. 5ta. Edición. Pearson Educación S.A. España.

- Tanenbaum, A. S. (1996). *Sistemas Operativos Distribuidos*. Prentice-Hall Hispanoamericana S.A. México.
- Tanenbaum, A. S. (2009). *Sistemas Operativos Modernos*. 3ra. Edición. Pearson Educación S. A. México.
- Tanenbaum, A. S., Van Steen, M. (2008). *Sistemas Distribuidos – Principios y Paradigmas*. 2da. Edición. Pearson Educación S. A. México.
- Tenenbaum, J. B., Griffiths, T. L. and Kemp, C. (2006). *Theory-Based Bayesian Models Of Inductive Learning And Reasoning*. Elsevier. Special Issue: Probabilistic Models Of Cognition. *TRENDS in Cognitive Sciences*. Vol.10, no. 7. USA.
- Varela, F. J. (1990). *Conocer - Las Ciencias Cognitivas: Tendencias y Perspectivas - Cartografía de las Ideas Actuales*. Gedisa Editorial. España.
- Wiener, N. (1961). *Cybernetics*, The M.I.T Press: Cambridge, Massachusetts, 2nd ed.
- Wiener, N. (1985). *Cibernética o el Control y la Comunicación en Animales y Máquinas*. 2ª Edición. Tusquets. España.
- Yager, R. (1988). *On Ordered Weighted Averaging Aggregation Operators In Multi-Criteria Decision Making*. *IEEE Trans. On Systems, Man and Cybernetics*. Vol. 18, pp. 183-190.
- Yager, R. (1993). *Families Of OWA Operators*. *Fuzzy Sets and Systems*. Vol. 59, pp. 125-148.
- Yager, R., Kacprzyk, J. (1997). *The Ordered Weighted Averaging Operators. Theory And Applications*. Kluwer Academic Publishers. USA.
- 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.

- Ynoub, R. (2007). El Proyecto Y La Metodología De Investigación. Ed. CENGAGE. Argentina.

Apéndice de publicaciones

- 1) D. L. la Red Martínez, F. Agostini (2017). "Modelo de Asignación de Recursos para la Enseñanza de los Procesos Distribuidos". Primer Congreso Latinoamericano de Ingeniería - CLADI. Centro Provincial de Convenciones Paraná, Entre Ríos, Argentina. ISBN 978-987-1896-84-4.
- 2) D. L. La Red Martínez, F. Agostini, C. Primorac (2018). "Nueva propuesta para la administración de recursos y procesos en sistemas distribuidos". Workshop de Investigadores en Ciencias de la Computación, WICC. ISBN 978-987-3619-27-4. Corrientes, Argentina.
- 3) D. L. La Red Martínez, J. C. Acosta, F. Agostini (2018). "Assignment of Resources in Distributed Systems". 9th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2018), Orlando, USA.
- 4) 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). The Science and Information (SAI) Organization, England, U.K. Vol. 9, no. 12.
- 5) Agostini F., La Red Martínez D. L., Acosta J. C., (2019). "Assignment of Resources in Distributed Systems With Strict Consensus Requirements". 10th International Multi-Conference on Complexity, Informatics and Cybernetics (IMCIC 2019), Orlando, USA.
- 6) Agostini F., La Red Martínez D. L., (2019). "Allocation of shared resources". 14^a Conferencia Ibérica de Sistemas y Tecnologías de Información (CISTI'2019). ISBN N° 978-989-98434-9-3, pp. 1-6. Universidad de Coimbra, Coimbra, Portugal. (Publicado).
- 7) Agostini F., Fornerón Martínez J. T., La Red Martínez D. L., (2019). "*Nuevo operador de agregación para grupos de procesos*". 16^a Conferencia Ibero Americana. Lisboa, Portugal.