

9° Congreso Nacional

# CoNaIISI 2021

de Ingeniería Informática  
y Sistemas de Información

4 **NOVIEMBRE**  
5



Departamento de Ingeniería  
en Sistemas de Información





**9° Congreso Nacional de  
Ingeniería Informática/Sistemas de Información  
CoNallSI  
4 y 5 de noviembre de 2021**

**Universidad Tecnológica Nacional  
Facultad Regional Mendoza**

**Memoria de Trabajos**

# Imputación de valores faltantes sobre la información de control en el contexto de los sistemas distribuidos

Diego David Duré Attis  
Facultad de Ciencias  
Aplicadas.  
Universidad Nacional  
de Pilar.  
Pilar, Paraguay  
[didadua@gmail.com](mailto:didadua@gmail.com)

Federico Agostini  
Facultad de Cs. Ex. y  
Naturales y Agrimensura.  
Universidad Nacional del  
Nordeste.  
Corrientes, Argentina  
[fagostini@exa.unne.edu.ar](mailto:fagostini@exa.unne.edu.ar)

David Luis La Red Martínez  
Facultad de Cs. Ex. y  
Naturales y Agrimensura.  
Universidad Nacional del  
Nordeste.  
Corrientes, Argentina  
[lrmdavid@exa.unne.edu.ar](mailto:lrmdavid@exa.unne.edu.ar)

## Resumen

*Los sistemas distribuidos están compuestos por varios ordenadores (nodos), físicamente separados pero conectados entre ellos por medio de una red de comunicaciones. Para esta propuesta existe un nodo central (ordenador principal) que es el encargado de administrar la información de control de todos los nodos, específicamente recibe y actualiza permanentemente la información de control de los nodos, para luego asignar los recursos asegurando la disponibilidad de los mismos en la modalidad de exclusión mutua y respetando las prioridades de los diferentes procesos. Para un ciclo de recolección de información de gestión, el nodo central puede recibir de alguno de los nodos información de control incompleta, lo cual es un inconveniente para la gestión de recursos y procesos. La propuesta para solucionar la problemática de estos valores faltantes sobre la información de control en los sistemas distribuidos, es aplicar el método de imputación K-Means, incorporando una capa de imputación o asignación a un modelo de decisión, que permitirá completar los valores faltantes con valores estimados, necesarios para establecer un correcto orden de asignación de recursos a los procesos. El K-Means es considerado uno de los métodos más fiables y de menor consumo de recursos computacionales, y uno de los más utilizados en investigaciones de imputación.*

## 1. Introducción

Los sistemas distribuidos son un conjunto de ordenadores que trabajan de forma coordinada, cuyos componentes (hardware y software) están conectados entre sí por una red de computadoras, e interactúan entre ellos para lograr un objetivo en común. Debido a la existencia de múltiples procesos que cooperan para el logro de una determinada función, se ve la necesidad de disponer de

modelos de decisión que permitan a los procesos intervinientes en los distintos grupos de procesos acceder a los recursos compartidos asegurando la disponibilidad de los mismos en la modalidad de exclusión mutua.

En los diferentes nodos se encuentran distribuidos múltiples procesos, en los cuales 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 de acuerdo a las características de los requerimientos de los procesos. 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.

El problema se presenta cuando la información de control respecto de la carga de los nodos llega al nodo central de manera incompleta, por ello, el nodo es considerado inaccesible, debido a que no puede brindar dicha información al runtime central (coordinador), a consecuencia de esto se van postergando innecesariamente solicitudes de recursos por parte de los procesos a siguientes rondas de asignación.

Las principales fuentes de interés se mencionan a continuación.

Algunos modelos de investigaciones sobre comunicación en sistemas distribuidos se presentan en [1], [2], en [3] y [4] se describen los principales algoritmos de comunicación en sistemas distribuidos (algoritmos clásicos de las ciencias de la computación).

Investigaciones relacionadas a sincronización en sistemas distribuidos se presentan en [5] donde se propone 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 [6], [7] y en [8] donde se presentan unos algoritmos para gestionar la exclusión mutua en redes de computadoras, conforme a las ciencias

de la computación, en [9] 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.

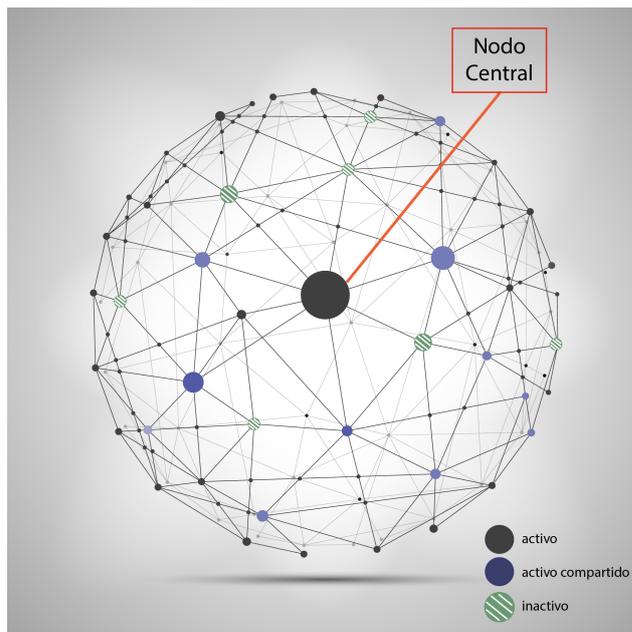
Los métodos considerados innovadores para la asignación de recursos en sistemas distribuidos respetando la exclusión mutua en el acceso a los recursos compartidos expresados en [10], [11], [12] donde se desarrollan operadores de agregación para asignaciones de recursos en sistemas distribuidos, en [13], [14] y [15] se presentan modelos de decisión innovadores para la gestión de recursos en sistemas distribuidos.

Los principales métodos de imputación se describen en [16], [17], [18], [19], [20], [21], [22], [23] y [24]. Los modelos de decisión utilizando operadores de agregación adecuados se muestran en [25], [26] y [27]. Los mecanismos de amputación que generan conjuntos de datos con valores faltantes a partir de conjuntos de datos completos se expresan en [28], [29] y [24].

Para solucionar la problemática se presenta una propuesta innovadora partiendo como punto de inicio de las investigaciones en [10] y [14], donde no consideran la posibilidad de información de control faltante en el desarrollo de los operadores de agregación para asignar recursos en sistemas distribuidos. Se pretende agregar a un modelo de decisión una capa de imputación/asignación, para que los valores faltantes puedan ser sustituidos por los valores estimados por el método seleccionado, utilizando valores históricos de ciclo de recolección y características conocidas de los nodos, para que finalmente el modelo de decisión pueda establecer un correcto orden de asignación de recursos. Serán consideradas las premisas y las estructuras de datos mencionadas en esas publicaciones.

## 2. Propuesta de Solución

En una macro imagen del sistema, se consideran transiciones de estados de acuerdo a la actualización iterativa de los requerimientos del sistema, en donde se encuentran los distintos nodos, cuyo estado individual puede ser: a) activo, cuando interactúa con los demás nodos aunque no envíe información ni participe del uso de recursos o la ejecución de procesos con recursos externos; b) activo compartido, cuando interactúa con los demás nodos, enviando y recibiendo información, creando procesos que utilizan recursos de otros nodos y compartiendo recursos propios con nodos externos; c) inactivo, cuando no se tiene información del estado real del nodo (puede estar sobrecargado y no puede realizar el intercambio de información, o bien, pudo haberse cortado el enlace que lo conecta con los demás nodos). Por último, existe un nodo central, que a través de un runtime gestiona la información de todos los nodos que participan en las asignaciones de recursos (ver **Figura 1**).



**Figura 1. Estados de los nodos.**

El nodo central, es el encargado de recopilar la información de todos los nodos, aplicar el proceso de agregación y obtener la lista de asignaciones de recursos a procesos. Un macro ciclo contempla la iteración de este proceso, en el cual el modelo de decisión es una etapa que se repite  $n$  veces. En este macro ciclo hay un paso previo de recopilación de información del estado del sistema, nodos activos y conectados, cuántos están entregando información, es decir un sondeo previo. Luego de la recepción se continúa con el proceso de evaluación y la emisión de los resultados. Se vuelve a aplicar el ciclo y el universo va cambiando en cada iteración. La falta de información en alguno de estos aspectos, será resuelta mediante los métodos comentados a continuación.

### 2.1. Método de imputación K-Means

Es un algoritmo de clasificación no supervisada que agrupa a las observaciones de forma tal que todas las que se encuentren en el mismo grupo sean lo más semejantes entre sí, la semejanza se calcula con una métrica de similitud, calculando la media del grupo seleccionado por la semejanza (imputación con la media), eso evita el sesgo en la varianza y en la desviación estándar (estimación en base a los valores parecidos y no al total). En K-Means la cantidad de valores parecidos surge de la clusterización realizada (si se tiene que imputar el valor de una variable de un registro de un grupo de 100 registros que componen el clúster, se tomarán los valores disponibles de esa variable a imputar dentro del grupo de los 100).

### 2.2. ¿Cómo funciona el algoritmo K-Means?

Según lo expuesto en [30], el algoritmo K-Means tiene como base la optimización de una función criterio, donde denomina  $F$ , el valor de esta función depende de las particiones del conjunto de datos  $\{C_1, \dots, C_k\}$

$$F : P_k(X) \Rightarrow R \quad (1)$$

Dónde:  $P_k(X)$ , son las particiones del conjunto de datos  $X = \{x_1, \dots, x_n\}$  en  $k$  grupos no vacíos,  $x_i$ , es un vector de  $n$ -dimensional (objeto) del conjunto de datos  $X$ .

El algoritmo K-Means converge a un mínimo local, utilizando la función de criterio  $F$ , de la sumatoria de distancias  $L2$  entre cada objeto y su centroide más cercano. A este criterio normalmente se le denomina error cuadrático y se obtiene a través de la expresión 2.

$$F(\{C_1, \dots, C_k\}) = \sum_{i=1}^k \sum_{j=1}^{p_i} \|x_{ij} - \bar{C}_i\| \quad (2)$$

Dónde:  $K$  es el número de grupos,  $p_i$  es el número de objetivos de grupo  $i$ ,  $x_{ij}$  es el  $j$ -ésimo objeto del  $i$ -ésimo grupo y  $\bar{C}_i$  es el centroide del  $i$ -ésimo grupo el cual es calculado a través de la expresión 3.

$$\bar{C}_i = \frac{1}{p_i} \sum_{j=1}^{p_i} x_{ij}, i = 1, \dots, K \quad (3)$$

El conjunto de pasos lógicos del algoritmo K-Means es el que se presenta a continuación:

**Paso 1.** Selecciona los  $K$  centroides iniciales  $\{C_1, \dots, C_k\}$ .

**Paso 2.** Asigna los objetivos  $x_i$  del conjunto de datos  $X$ , a su centroide más cercano.

**Paso 3.** Recalcula los nuevos centros, regresa al paso 2, hasta que el algoritmo converge.

El algoritmo se inicia seleccionando o calculando los centroides iniciales, dependiendo del criterio de selección de centroides, posteriormente asigna los objetos a su centroide más cercano, para después recalcular los nuevos centroides esto lo realiza hasta que el algoritmo converja (paso 3).

Pre procesados los datos y aplicada la técnica de limpieza de datos, los datos faltantes se rellenan, utilizando el método de imputación por media condicional. Método que sustituye los valores faltantes de una variable mediante la media de las unidades observadas en esa variable.

### 2.3. Amputación de datos

En esta propuesta, se realiza la amputación con el objetivo de generar datos faltantes en un archivo histórico de información de control, para luego imputar los datos de control cuyo faltante se ha generado, para poder comparar los resultados con datos completos (estos datos son los utilizados en [10], específicamente en el ejemplo donde se presentan las valoraciones de los criterios establecidos para la carga computacional actual de los nodos y las prioridades o preferencias de los procesos) con los resultados logrados con la imputación propuesta.

La amputación es el proceso por el cual se generan conjuntos de datos con valores faltantes a partir de conjuntos de datos completos [29].

En [24] se generaron conjuntos de datos con valores faltantes a partir de un conjunto de datos completo, para luego imputarlos mediante diferentes métodos de imputación. El entorno de trabajo desarrollado para realizar los experimentos de amputación y posterior imputación resulta apropiado y permite la incorporación a futuro de

otros mecanismos de amputación y otros métodos de imputación.

Existen diferentes mecanismos para generar la pérdida de datos, [31] clasificó los mecanismos en tres tipos diferentes:

- Falta completamente aleatoria (**MCAR**: missing completely at random): representa una situación para la cual la ausencia es independientemente de las variables. Es una ausencia de datos debida exclusivamente al azar.
- Falta aleatoria (**MAR**: missing at random): se refiere cuando la ausencia de datos está ligada a las variables independientes del estudio, pero no a la variable dependiente.
- Falta no aleatoria (**NMAR**: not missing at random): estos casos se dan cuando la pérdida de datos se debe a la variable dependiente, y posiblemente a alguna variable independiente.

### 2.4. Aplicar Mecanismos de amputación con el lenguaje de programación R

Sin pérdida de generalidad, para la amputación de datos se utiliza el lenguaje de programación R [32]. Para obtener los valores perdidos se adopta el método de amputación multivariado descrito en [29] implementado en la función "ampute" del paquete "mice" [33] del software R [34]. Para generar un escenario de valores faltantes se procede a realizar la amputación de datos, seleccionando para este ejemplo el mecanismo MCAR, parametrizado en un 10%.

### 2.5. Aplicar el método K-Means con el lenguaje de programación R

Sin pérdida de generalidad, para la imputación de datos en esta propuesta se utiliza el lenguaje de programación R [34]. El método de imputación seleccionado es el K-Means, éste imputa el valor faltante utilizando el lenguaje de programación mencionado, primeramente, calcula la cantidad óptima de clúster, y seguidamente utiliza la información de los clúster obtenidos para imputar los valores faltantes.

## 3. Operador de agregación utilizado por el Modelo de Decisión

En [10], las características de los operadores de agregación descritos permiten considerar que el método propuesto pertenece a la familia de operadores de agregación Neat-OWA. La propuesta desarrollada en el trabajo citado consiste en generar un modelo de decisión y su correspondiente operador de agregación para la gestión de grupos de procesos que comparten recursos, realizando modificaciones de los operadores de la familia OWA mencionados.

La propuesta mencionada anteriormente no considera la información de control faltante, por lo que en este trabajo se propone incorporar una capa de imputación/asignación (ver **Figura 2**), para imputar los valores de las variables de

categorías de carga computacional y otra para los valores de las variables de criterios de carga computacional, teniendo en cuenta los siguientes pasos: **a)** Identificar el propósito del problema; **b)** Identificar las alternativas; **c)** Listar los criterios que van a ser tenidos en cuenta para elegir la mejor alternativa.



Figura 2. Capas de imputación/asignación en el Modelo de Decisión.

### 3.1. Identificar el propósito del problema

En esta investigación se pretende resolver las solicitudes de los diferentes recursos por parte de los procesos ubicados en los nodos del sistema distribuido. La información de control compartida por los distintos nodos es recibida por el runtime alojado en el nodo central, y a través de un operador de agregación determina y soluciona la siguiente premisa: que los procesos accedan a recursos compartidos en la modalidad de exclusión mutua, incorporando mecanismos de imputación de datos para los casos en que la información sobre las variables que indican el estado de carga de alguno o algunos de los nodos sea incompleta.

En un ciclo de recolección de información el nodo central puede recibir de alguno de los nodos información incompleta; por ejemplo, alguno de los criterios utilizados para calcular la carga computacional o las prioridades o preferencia de los procesos. Para solucionar la problemática planteada se desea mejorar un modelo de decisión y su correspondiente operador de agregación incorporando una capa de imputación/asignación, para que el mecanismo pueda completar los valores faltantes de la información de control, y finalmente el modelo de decisión pueda establecer un orden de asignación de recursos. En caso de que la información suministrada por los nodos esté completa, el modelo de decisión se ejecuta sin necesidad de la imputación de datos, según la propuesta de [10] y [14].

En base a revisiones bibliográficas y por ser uno de los métodos más fiables y de menor consumo de recursos computacionales, se opta por utilizar el método de imputación K-Means, y considerando que es uno de los métodos más utilizados, en razón de sus resultados, en investigaciones donde los resultados obtenidos son comparados con otros métodos, como en las propuestas de [18] y [24].

### 3.2. Identificar las alternativas

Son las opciones que podrían ayudar a la consecución del propósito que se trazó en el primer paso. En cada nodo

se define una interfaz entre las aplicaciones y el sistema operativo, que a través de un runtime (software en tiempo de ejecución complementario al sistema operativo) incluido en esa interfaz, gestiona los procesos y recursos compartidos y define el escenario correspondiente.

En un determinado ciclo de recolección de información proporcionada por todos los nodos, el nodo central puede recibir información incompleta de uno o algunos de los criterios utilizados para el cálculo de la carga computacional de cada nodo.

Además, para cada nodo, se consideran distintos criterios para calcular la prioridad o preferencia que tiene cada uno, en las asignaciones de recursos a procesos. Al igual que el caso anterior, algunos de los valores de estos criterios podrían no estar disponibles en un ciclo de recolección de información de control.

### 3.3. Listar los criterios que van a ser tenidos en cuenta para elegir la mejor alternativa

#### 3.3.1. Indicadores de las prestaciones actuales de cada nodo

Para obtener un indicador de las prestaciones actuales de cada nodo se adoptarán las  $j$  prestaciones en los  $n$  nodos. El runtime del nodo que arranca estará informando al runtime del nodo central sus características, por ejemplo, la velocidad de procesamiento, capacidad de memoria, velocidad de transmisión de datos, velocidad de entrada de salida, entre otros.

Establecimiento de las prestaciones para todos los nodos.

$prestaciones = \{pres_{ij}\}$  con  $i = 1, \dots, n$  (nº de nodos en el sistema distribuido) y  $j = 1, \dots, s$  (prestaciones para cada nodo).

#### 3.3.2. 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  $e$  criterios en los  $n$  nodos. Los valores que se asumirán para los indicadores de carga computacional de los  $n$  nodos y el cálculo de carga promedio para cada nodo serán los obtenidos por el nodo central.

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

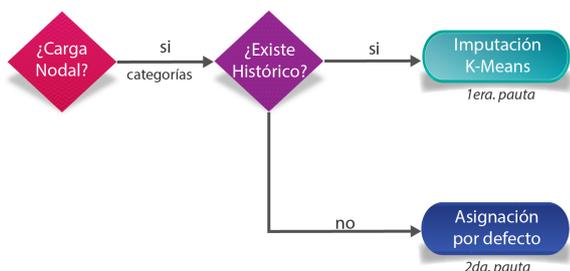
Cálculo de la carga computacional de cada nodo:

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

Puede presentarse la situación de que en cierto ciclo de recolección el nodo central reciba información incompleta de uno o algunos de los criterios (% de uso de CPU, % de uso de Memoria y/o el % de uso de Operación de Entrada/Salida) a utilizar para el cálculo de la carga computacional, entonces se aplica la imputación de datos para dichos criterios, para imputar se tendrá en cuenta

ciertas pautas, sabiendo que son las variables de estados consideradas para los nodos.

En función a la información de control histórica (disponible en el nodo central y que corresponde a la información de carga computacional y preferencias de los procesos en ciclos anteriores) disponible para los métodos de imputación/asignación, se describirán distintas pautas como se puede observar a continuación (ver **Figura 3**).



**Figura 3. Diagrama de flujos para carga nodal.**

La carga nodal se refiere a las variables que indican el estado de carga de los nodos.

Analizando la **Figura 1** se establecen las siguientes pautas para resolver el problema del valor faltante:

Primera pauta; se analiza si se cuenta con información histórica del nodo en cuestión, de ser afirmativo se **imputa con el método K-Means**, considerando una cantidad de registros históricos de más de diez ciclos de recolección.

Segunda pauta; si no se cuenta con información histórica del nodo, se realizará la **asignación de un valor por defecto**, en función de la carga computacional informada y las características del nodo.

Para la **carga computacional informada** se calcula el promedio de la carga computacional existente en base a la información disponible de los valores de criterios, que es igual a la suma de los valores disponibles de criterios de carga computacional dividido la cantidad de valores disponibles.

En base a las **características del nodo** informadas previamente, se asume un valor, de acuerdo a sus prestaciones (un valor distinto para los nodos de Alta, Media y Bajas prestaciones).

Se promedia los valores obtenidos de la **carga computacional informada** y las **características del nodo**, sumando ambos valores y el resultado, dividiéndolo por dos. De esta manera se obtiene el valor por defecto como lo establece la segunda pauta.

Mediante la imputación o asignación de valores faltantes se obtiene los valores faltantes de los criterios en cuestión, permitiendo de esta manera obtener los indicadores de carga computacional actual de cada nodo como se indica inicialmente.

### 3.3.3. El establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a ellos

En esta propuesta, serán utilizados los mismos criterios que en [10] y [14].

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:

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

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

$\text{categorías} = \{\text{cat}_{ij}\}$  con  $i = 1, \dots, n$  (n° de nodos en el sistema distribuido) y  $j = 1, \dots, a$  (n° máximo de categorías para cada nodo).

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.

En esta propuesta se hará una clasificación de los criterios, que serán agrupados de acuerdo a:

- 1) Criterios relacionados con las variables de estados principales del nodo.
- 2) Criterios referentes a las prioridades de proceso.
- 3) Criterios que hacen referencia a las sobrecargas.

Sin embargo, esta clasificación podría variar para cada nodo y para cada circunstancia.

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:

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

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

$\text{criterios para preferencias} = \{\text{cp}_{isj}\}$  con  $i = 1, \dots, a$  (n° de categorías de carga computacional),  $s = 1, \dots, t$  (n° de

clasificación de criterios) y  $j = 1, \dots, e$  (n° máximo de criterios)

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):

$pesos = \{w_{isj}\}$  con  $i = 1, \dots, a$  (n° de categorías de carga computacional),  $s = 1, \dots, t$  (n° de clasificación de criterios) y  $j = 1, \dots, e$  (n° máximo de criterios).

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:

$w_{isj} = norm(función(cp_{isj}))$  con  $i = 1, \dots, a$  (n° de categorías de carga computacional),  $s = 1, \dots, t$  (n° de clasificación de criterios) y  $j = 1, \dots, e$  (n° máximo 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:

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

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.

### 3.3.4. Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo

Las prioridades o preferencias se calculan en cada nodo para cada solicitud de recursos originada en cada proceso, se establecen los criterios que se aplicarán (iguales para todos los nodos y que pueden pertenecer a una sola clasificación):

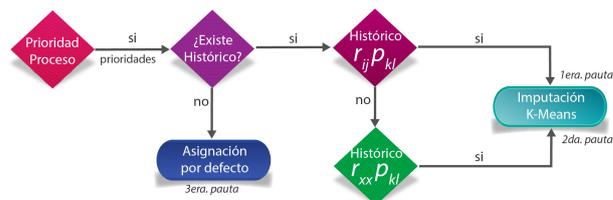
$valoraciones (r_{ij} p_{kl}) = \{cp_m\}$  con  $i = 1, \dots, n$  (nodo donde reside el recurso),  $j = 1, \dots, r$  (recurso en el nodo  $i$ ),  $k = 1, \dots, n$  (nodo donde reside el proceso),  $l = 1, \dots, p$  (proceso en el nodo  $k$ ),  $m = 1, \dots, e$  (valoración de criterios) y  $s = 1, \dots, t$  (n° de clasificación de los criterios).

Los valores de los criterios de evaluación de carga permiten calcular las prioridades o preferencias de los procesos, el problema que podría presentarse durante un ciclo de recolección de información de control, es que el nodo central reciba de alguno o algunos nodos información incompleta de los mismos, entonces se aplica el método de imputación para completar los valores faltantes en el momento, y así resolver la problemática mencionada.

Los distintos criterios establecidos para la prioridad de proceso ayudan para calcular la prioridad o preferencia que tiene cada proceso en las asignaciones de recursos, de ahí la importancia de la disponibilidad de los valores de estos

criterios en cada ciclo de recolección. Se tendrán en cuenta para las variables, tres clasificaciones establecidas con sus respectivas pautas, que en función a la información de control histórica disponible para los métodos de imputación/asignación, se describirán distintas pautas para las diferentes clasificaciones.

*Clasificación 1:* incluye variables medidas por el sistema operativo que representan la carga del nodo, por ejemplo, las variables de estados principales del nodo, que son las medidas que toma el sistema operativo cada vez que le llega un requerimiento. Las pautas establecidas se describen a continuación (ver **Figura 4**).



**Figura 4. Diagrama de flujos para la prioridad o preferencia de procesos (clasificación 1)**

Primera pauta: se analiza si se cuenta con información histórica, y si corresponde a la misma relación recurso-proceso  $r_{ij} p_{kl}$ , de ser afirmativo se imputa con el método K-Means, considerando un histórico de valores superior a diez ciclos de recolección de información.

Segunda pauta: se analiza si se cuenta con información histórica, pero no se tiene específicamente de la relación recurso-proceso  $r_{ij} p_{kl}$ , sino se tiene información histórica del proceso con relación a otros recursos  $r_{xx} p_{kl}$ , se aplica la imputación de datos con K-Means, teniendo en cuenta un histórico de valores superior a diez ciclos de recolección de información.

Tercera pauta: si no se cuenta con información histórica específica de la relación recurso-proceso  $r_{ij} p_{kl}$ , tampoco se tiene información histórica del proceso con relación a otros recursos  $r_{xx} p_{kl}$ , se realizará la asignación de un valor por defecto, en función de la **carga computacional informada** y las **características del nodo**.

Se promedia los valores obtenidos de la **carga computacional informada** y las **características del nodo**, sumando ambos valores y el resultado dividiendo en dos.

De esta manera se obtiene el valor por defecto como lo establece la tercera pauta.

*Clasificación 2:* son los valores asignados por defecto por el sistema operativo cuando se genera el proceso, por ejemplo, la Prioridad Proceso. Las pautas establecidas se describen a continuación (ver **Figura 5**).

Primera pauta: para **a** se analiza si se cuenta con información histórica en **b**, y si es de la misma relación recurso-proceso  $r_{ij} p_{kl}$  en **c**, de ser afirmativo se imputa con el método K-Means en **d**, considerando un histórico de valores superior a diez ciclos de recolección de información.

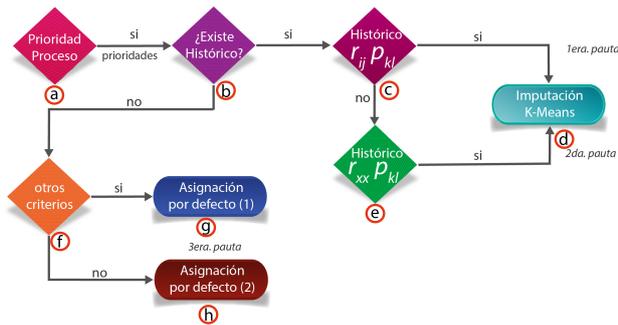


Figura 5. Diagrama de flujos para la prioridad o preferencia de procesos (clasificación 2).

Segunda pauta: para **a** se analiza si se cuenta con información histórica en **b**, pero no se tiene específicamente de la relación recurso-proceso  $r_{ij} p_{kl}$  en **c**, sino se tiene información histórica del proceso con relación a otros recursos  $r_{xx} p_{kl}$  en **e**, se aplica la imputación de datos con K-Means en **d**, teniendo en cuenta un histórico de valores superior a diez ciclos de recolección de información.

Tercera pauta: para **a** si no se cuenta con información histórica en **b** (específicamente de la relación recurso-proceso  $r_{ij} p_{kl}$  en **c**, tampoco se tiene información histórica del proceso con relación a otros recursos  $r_{xx} p_{kl}$  en **e**) y al no tener información sobre otros criterios referente a la clasificación 2 en **f**, se asumirá un **valor por defecto (2)** (se considera cuando se tiene un solo criterio para la clasificación 2), en función a las características del nodo.

De esta manera se obtiene el valor por defecto como lo establece la tercera pauta.

Cuarta pauta: para **a** si no se cuenta con información histórica en **b** (específicamente de la relación recurso-proceso  $r_{ij} p_{kl}$  en **c**, tampoco se tiene información histórica del proceso con relación a otros recursos  $r_{xx} p_{kl}$  en **e**), y en caso de tener información sobre otros criterios referente a la clasificación 2 en **f**, se realizará la asignación de un **valor por defecto (1)** (se considera cuando se tiene más de un criterio para la clasificación 2), en función de la carga computacional informada y las características del nodo.

Se promedia los valores obtenidos de la **carga computacional informada** y las **características del nodo**, sumando ambos valores y el resultado dividiendo en dos.

De esta manera se obtiene el valor por defecto como lo establece la cuarta pauta.

**Clasificación 3:** son los valores que indican cuánto costaría al nodo hacer la asignación, el impacto esperado de atender el requerimiento (se tiene en cuenta la relación de proceso-recurso). Las pautas establecidas se describen a continuación (ver **Figura 6**).

Primera pauta: se analiza si se cuenta con información histórica, y si corresponde a la misma relación recurso-proceso  $r_{ij} p_{kl}$ , de ser afirmativo se imputa con el método K-Means, considerando un histórico de valores superior a diez ciclos de recolección de información.

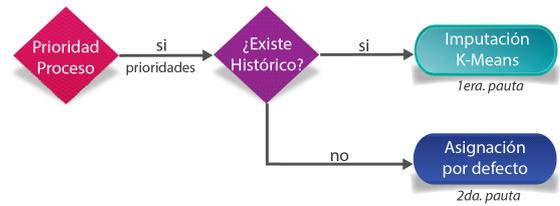


Figura 6. Diagrama de flujos para la prioridad o preferencia de procesos (clasificación 3).

Segunda pauta: si no se cuenta con información histórica específica de la relación recurso-proceso  $r_{ij} p_{kl}$ , y al no considerar para esta clasificación información histórica del proceso con relación a otros recursos  $p_{kl} r_{xx}$ , se realizará la **asignación de un valor por defecto**, en función de la carga computacional informada y las características del nodo.

Se promedia los valores obtenidos de la **carga computacional informada** y las **características del nodo**, sumando ambos valores y el resultado dividiendo en dos.

De esta manera se obtiene el valor por defecto como lo establece la segunda pauta.

Luego de aplicar la imputación de datos faltantes necesarios para tener toda la información que utiliza el modelo de decisión para resolver los distintos escenarios desarrollados en [10] se prosigue con los cálculos previstos en el mismo.

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

Los vectores de 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  $k$ ),  $m = 1, \dots, e$  (criterios de valoración de la prioridad del requerimiento) y  $s = 1, \dots, t$  ( $n^\circ$  de clasificación de criterios).

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

### 3.3.5. Cálculo de las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles

Las prioridades de los procesos se las calculan en el administrador centralizado de recursos compartidos, para que seguidamente se determine el 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 colocan las prioridades o preferencias nodales calculadas en la etapa anterior; donde cada fila contiene la información de las prioridades nodales de los distintos procesos para acceder a un determinado recurso.

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), 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).

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

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 tomadas fila por fila; es decir, respecto de cada recurso, se multiplicarán escalarmente por el vector de pesos finales normalizados 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*):

$FASD(r_{ij}) = \sum nwf_{kl} * p_{kl}$  = 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.

## 4. Ejemplo

Se analizará detalladamente un ejemplo con la aplicación del operador de agregación propuesto. El sistema de procesamiento distribuido, las estructuras de datos, los recursos y los procesos que se ejecutan en los diferentes nodos, grupos, cardinalidades, criterios y categorías para evaluar las diferentes cargas y cálculos necesarios, son los mencionados en [10].

### 4.1. Identificar el propósito del problema

El runtime alojado en el nodo central se encarga de recibir y mantener actualizada la información de control de todos los nodos, para luego realizar la asignación de recursos a procesos en la modalidad de exclusión mutua. Para este ejemplo, en un ciclo de recolección de información de gestión, necesaria para ejecutar y asegurar lo mencionado anteriormente, el nodo central recibe de algunos de los nodos información incompleta de los criterios de evaluación de carga nodal y de las prioridades o preferencias de los procesos para cada nodo.

La falta de información en algunos de estos aspectos, será resuelta mediante la incorporación de una capa de imputación/asignación, a un modelo de decisión existente para gestión de recursos y procesos en sistemas distribuidos, considerando como punto de inicio las premisas y las estructuras de datos mencionadas en [10] y [14].

### 4.2. Identificar las alternativas

El Runtime del nodo central recibe información incompleta de algunos de los criterios utilizados, tanto para calcular la carga computacional de cada nodo, y/o para calcular las prioridades o preferencias de los procesos para cada nodo, se consideran diferentes pautas para completar dichos valores, utilizando métodos de imputación o asignación.

En situaciones donde falta la información de algunos de los criterios para calcular la carga computacional de cada nodo, se considera la imputación si se tiene un histórico de información respecto del nodo donde se encuentra el dato faltante, de no ser así, se hace la asignación en función a la carga computacional informada, y las características del nodo.

Al igual que la sección anterior, se consideran distintos criterios para calcular la prioridad o preferencia de los procesos para cada nodo, algunos de los valores de la información de control de estos criterios pueden no estar disponibles. Se aplica el método de imputación o asignación para obtener los valores que sustituyen a los valores faltantes, dependiendo de la clasificación a la que corresponda el criterio afectado con valor faltante.

Los runtime de cada nodo informan al runtime del nodo central sus características, y así se obtiene un indicador de las prestaciones los mismos.

En caso de que la información suministrada por los nodos esté completa, el modelo de decisión se ejecuta sin necesidad de la imputación de datos, según la propuesta de [10] y [14].

### 4.3. Listar los criterios que van a ser tenidos en cuenta a elegir la mejor alternativa

#### 4.3.1. Indicadores de las prestaciones de cada nodo

Para obtener un indicador de las prestaciones del nodo, se tienen en cuenta sus características, por ejemplo, la velocidad de procesamiento, capacidad de memoria, velocidad de transmisión de datos, velocidad de entrada de salida, entre otros. El runtime del nodo que arranca está informando al runtime del nodo central sus características (ver **Tabla 1**). Los detalles de implementación de las mismas son:

- Nodo de prestaciones Altas se asume un valor de 30%.
- Nodo de prestaciones Medias se asume un valor de 50%.
- Nodo de prestaciones Bajas se asume un valor 70%.

Las prestaciones correspondientes a cada nodo se muestran en la **Tabla 1**.

**Tabla 1. Valores de las características referentes a las prestaciones de cada nodo**

Nodos	Prestaciones
1	Bajas
2	Medias
3	Altas

#### 4.3.2. Cálculo de la carga computacional actual de los nodos

En un ciclo de recolección de información proporcionada por todos los nodos, el nodo central puede recibir información incompleta de algunos de los criterios utilizados para calcular la carga computacional de cada nodo. Teniendo en cuenta que en [10] y [14] no se consideran valores faltantes en la información de los criterios de carga computacional, entonces, para generar un escenario de valores faltantes para este ejemplo, se procede a realizar la amputación de datos, que según [29] es el proceso por el cual se generan conjuntos de datos con valores faltantes a partir de conjuntos de datos completos.

La amputación se hace para generar datos faltantes de información de control, para luego imputar y poder comparar los resultados logrados con la imputación propuesta. Considerando que el ejemplo tiene solo tres nodos, la información faltante de los distintos escenarios se ha generado amputando manualmente las valoraciones, tomando los datos del ejemplo con datos completos que figuran en la Tabla 17 de [10]. En una situación real, no se hace amputación, sólo imputación de datos faltantes.

Se observan valores faltantes en el % de Uso de CPU del nodo 1, % de Uso de Memoria del nodo 2 y el % de Uso de Oper. de E/S del nodo 3 (ver **Tabla 2**).

**Tabla 2. Valoraciones de los criterios de carga computacional con valor faltante**

Nodos	% Uso de CPU	% Uso de Memoria	% Uso de Oper. E/S
1	X	90	75
2	45	X	65
3	10	25	X

Nota: Valor faltante en los criterios del nodo 1, nodo 2 y nodo 3.

Para obtener la carga computacional actual de cada nodo se debe solucionar la problemática de valores faltantes mencionados en la **Tabla 2**. Para la solución de la misma se aplica el método de imputación/asignación teniendo en cuenta las pautas establecidas.

Después de haber analizado la información con que se cuenta sobre los criterios de carga computacional de los nodos donde existen valores faltantes; y teniendo en cuenta las pautas establecidas para la imputación o asignación de valores, queda clasificada de la siguiente manera (ver **Tabla 3**).

**Tabla 3. Clasificación de valores faltantes de los criterios de carga computacional de los nodos, de acuerdo a las pautas establecidas**

Valor Faltante	Pauta 1: Existe información histórica	Pauta 2: No existe información histórica
Nodo 1 – % Uso CPU	SI	
Nodo 2 – % Uso Memoria		SI
Nodo 3 – % Uso de Oper. E/S	SI	

- % Uso CPU – **Nodo 1**, % Uso de Oper. E/S – **Nodo 3**: como se muestra en la clasificación de la **Tabla 3**, para ambos casos en cuestión, se cuenta con información histórica respecto de la carga de los nodos, esto permite hacer la imputación con K-Means como lo establece la primera pauta. Para cada nodo, la tabla de valores históricos cuenta con mil registros, que han sido obtenidos desde un simulador (creado para este fin), que genera datos históricos de manera aleatoria sobre la carga computacional de cada nodo. Cuando un nodo ingresa por primera vez al sistema (el simulador crea un nuevo nodo), éste no tendrá información histórica, hasta que se realicen reiteradas simulaciones.

Aplicando la imputación de datos con un histórico de valores correspondientes a mil ciclos de recolección de información, se obtiene el valor faltante para cada criterio.

- **% Uso de Memoria – Nodo 2:** como se muestra en la clasificación de la **Tabla 3**, para este criterio de carga computacional del nodo 2, no se cuenta con información histórica, esto permite hacer la asignación de valor por defecto en función de la carga computacional informada y las características del nodo, como lo establece la segunda pauta.

Los valores que se asumirán para los indicadores de carga computacional de los tres nodos y el cálculo de carga promedio para cada nodo, posterior a la imputación/asignación se muestran a continuación (ver **Tabla 4**).

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

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

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

**Tabla 4. Valores de los criterios de carga computacional en cada nodo, con valoraciones imputados/asignado**

Nodos	Valores de los Criterios			Promedio
	% Uso CPU	% Uso Memoria	% Uso Oper. E/S	
1	75,69	90	75	80,23
2	45	52,5	65	54,16
3	10	25	69,47	34,82

*Nota:* Las celdas remarcadas son valores obtenidos mediante la imputación y/o asignación.

#### 4.3.3. Establecimiento de las categorías de carga computacional y de los vectores de pesos asociados a ellos

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%).

$$card(\{categorías\}) = 3.$$

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

Los valores obtenidos para las categorías de carga en base a los promedios indicados (ver **Tabla 4**), se observan a continuación (ver **Tabla 5**).

**Tabla 5. Valores de las categorías para medir la carga computacional en cada nodo**

Nodos	Valores de las categorías
1	Alta
2	Media
3	Baja

Para definir 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 se hará una clasificación de los siguientes criterios: 1) Clasificación 1 (N° de procesos en el nodo, % de uso de CPU, % de uso de memoria, % de uso de memoria virtual); 2) Clasificación 2 (prioridad del proceso); 3) Clasificación 3 (sobrecarga de memoria, sobrecarga de procesador y sobrecarga de entrada / salida).

Establecimiento del número 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:

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

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}.

A continuación, 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. Pueden observarse a continuación según su clasificación (ver **Tabla 6**, **Tabla 7** y **Tabla 8**).

**Tabla 6. Vectores de pesos para la clasificación 1**

Categorías	Clasificación 1			
	N° Proc.	% CPU	% Mem.	% MV
Alta	0,050	0,050	0,100	0,500
Media	0,100	0,200	0,300	0,100
Baja	0,100	0,300	0,200	0,200

**Tabla 7. Vectores de pesos para la clasificación 2**

Categorías	Clasificación 2
	Prioridad Proc.
Alta	0,100
Media	0,200
Baja	0,100

**Tabla 8. Vectores de pesos para la clasificación 3**

Categorías	Clasificación 3		
	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S
Alta	0,100	0,050	0,050
Media	0,050	0,025	0,025
Baja	0,025	0,025	0,050

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 (ver **Tabla 9**).

**Tabla 9. Sumatoria de pesos asignados por clasificación**

Categorías	Clasificaciones			$\sum w=1$
	$\sum$ Clasif. 1	$\sum$ Clasif. 2	$\sum$ Clasif. 3	
Alta	0,700	0,100	0,200	1
Media	0,700	0,200	0,100	1
Baja	0,800	0,100	0,100	1

#### 4.3.4. Cálculo de las prioridades o preferencias de los procesos teniendo en cuenta el estado del nodo

Para el mismo ciclo de recolección de información utilizado en la sección anterior se parte de datos y resultados de los paper de [10] y [14], donde no consideran valores faltantes en la información de los criterios establecidos para calcular las prioridades o preferencias de los procesos para cada nodo, entonces, a partir de esos datos completos se aplica la amputación para simular faltantes de datos de control, seleccionando para este ejemplo el mecanismo MCAR parametrizado en un 10%, finalmente imputarlos y verificar si los resultados obtenidos con los métodos de imputación/asignación funcionan adecuadamente. En una situación real, no se hace amputación, sólo imputación de datos faltantes.

Obtenidos los valores faltantes en algunos criterios, se aplica el método de imputación/asignación teniendo en cuenta la clasificación de los mismos y sus respectivas pautas.

- Clasificación 1 corresponde a las variables de estados principales del nodo, que son las medidas que toma el sistema operativo cada vez que le llega un requerimiento, seguidamente puede apreciarse las valoraciones faltantes (ver **Tabla 10**).

**Tabla 10. Valores faltantes para la clasificación 1**

Proceso Recurso	Clasificación 1			
	Nº Proc.	% CPU	% Mem.	% MV
$p_{11}r_{23}$	X	0,60	0,80	0,80
$p_{22}r_{22}$	X	0,90	0,60	0,80
$p_{25}r_{21}$	0,20	0,80	X	0,90
$p_{34}r_{33}$	0,40	0,60	0,80	X
$p_{35}r_{24}$	0,50	X	0,90	0,40

- Clasificación 2 corresponde a Prioridad de Proc., son valores asignados por defecto por el sistema operativo cuando se genera el proceso, a continuación puede apreciarse las valoraciones faltantes (ver **Tabla 11**).

**Tabla 11. Valores faltantes para la clasificación 2**

Proceso Recurso	Clasificación 2
	Prioridad Proc.
$p_{24}r_{12}$	X

- Clasificación 3, corresponden a los criterios de sobrecargas, que son los valores que indican cuánto costaría al nodo hacer la asignación, el impacto esperado de atender el requerimiento (se tiene en cuenta la relación de proceso-recurso), a continuación puede apreciarse las valoraciones faltantes (ver **Tabla 12**).

**Tabla 12. Valores faltantes para la clasificación 3**

Proceso Recurso	Clasificación 3		
	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S
$p_{12}r_{21}$	0,90	X	0,20
$p_{13}r_{11}$	0,50	X	0,80
$p_{23}r_{32}$	0,60	0,60	X
$p_{33}r_{13}$	0,40	0,50	X
$p_{33}r_{22}$	X	0,90	0,70
$p_{35}r_{12}$	0,50	0,70	X

Después de haber analizado las informaciones con que se cuenta sobre las variables donde faltan valores para las diferentes clasificaciones, y teniendo en cuenta las pautas para la imputación o asignación de valores, la relación de proceso-recurso queda clasificada de la siguiente manera (ver **Tabla 13**).

**Tabla 13. Clasificación de valores faltantes de acuerdo a las clasificaciones y pautas establecidas**

Valor Faltante	Clasific.	$r_{ij} p_{kl}$	$r_{xx} p_{kl}$	Asign.
$p_{11}r_{23}$ - Nº Proc.	1	SI		
$p_{22}r_{22}$ - Nº Proc.	1		SI	
$p_{25}r_{21}$ - % Mem.	1			SI
$p_{34}r_{33}$ - % MV	1		SI	
$p_{35}r_{24}$ - % CPU	1	SI		
$p_{24}r_{12}$ - Pri. Proc	2	SI		
$p_{12}r_{21}$ - Sobre. Proc.	3	SI		
$p_{13}r_{11}$ - Sobrec. Proc.	3	SI		
$p_{23}r_{32}$ - Sobrec. E/S	3	SI		
$p_{33}r_{13}$ - Sobrec. E/S	3	SI		
$p_{33}r_{22}$ - Sobrec. Mem.	3	SI		
$p_{35}r_{12}$ - Sobrec. Proc.	3			SI

*Gris claro*: indica que existe pauta para la columna de relación de procesos-recursos, pero no se considera porque ha sido seleccionada otra pauta previo análisis de registro histórico.

*Gris oscuro*: indica que no aplica ninguna pauta, para la columna de relación de procesos con otros recursos, para los criterios de la clasificación 3 (sobrecargas).

Para  $p_{11}r_{23}$ ,  $p_{35}r_{24}$ ,  $p_{24}r_{12}$ ,  $p_{12}r_{21}$ ,  $p_{13}r_{11}$ ,  $p_{23}r_{32}$ ,  $p_{33}r_{13}$  y  $p_{33}r_{22}$  se cuenta con información histórica de la misma relación de recurso-proceso  $r_{ij} p_{kl}$ , esto permite hacer la imputación con K-Means como lo establece la primera pauta, con valores históricos de ciclos de recolección para cada relación, obtenidos desde un simulador.

Para  $p_{22}r_{22}$  y  $p_{34}r_{33}$  no se cuenta con información histórica de la misma relación de recurso-proceso  $r_{ij} p_{kl}$ , pero se cuenta con información histórica de la relación de proceso con otros recursos  $r_{xx} p_{kl}$ , esto permite hacer la imputación con K-Means como lo establece la segunda pauta.

Para  $p_{25}r_{21}$  y  $p_{35}r_{12}$  no se cuenta con información histórica de la relación de proceso recurso, para  $p_{25}r_{21}$  tampoco se cuenta con información histórica de la relación de proceso con otros recursos  $r_{xx} p_{kl}$ , y el caso de  $p_{35}r_{12}$  no considera dicha información para la clasificación donde corresponde, esto permite hacer la asignación de valor por defecto en función de la carga computacional informada y las características del nodo.

Se obtienen los valores de los criterios mediante la imputación o asignación, que serán utilizados para calcular la prioridad o preferencia de procesos para cada nodo, se pueden observar a continuación (ver **Tabla 14**, **Tabla 15** y **Tabla 16**).

**Tabla 14. Valoraciones para la clasificación 1, con valores imputado/asignado**

Proceso Recurso	Clasificación 1			
	Nº Proc.	% CPU	% Mem.	% MV
$p_{11}r_{23}$	<b>0,38</b>	0,60	0,80	0,80
$p_{22}r_{22}$	<b>0,47</b>	0,90	0,60	0,80
$p_{25}r_{21}$	0,20	0,80	<b>0,55</b>	0,90
$p_{34}r_{33}$	0,40	0,60	0,80	<b>0,39</b>
$p_{35}r_{24}$	0,50	<b>0,46</b>	0,90	0,40

Nota: Los valores marcados son obtenidos mediante la imputación y/o asignación.

**Tabla 15. Valoraciones para la clasificación 2, con valor imputado**

Proceso Recurso	Clasificación 2
	Prioridad Proc.
$p_{24}r_{12}$	<b>0,62</b>

**Tabla 16. Valoraciones para la clasificación 3, con valores imputado/asignado**

Proceso Recurso	Clasificación 3		
	Sobrec. Mem.	Sobrec. Proc.	Sobrec. E/S
$p_{12}r_{21}$	0,90	<b>0,69</b>	0,20
$p_{13}r_{11}$	0,50	<b>0,30</b>	0,80
$p_{23}r_{32}$	0,60	0,60	<b>0,25</b>
$p_{33}r_{13}$	0,40	0,50	<b>0,77</b>
$p_{33}r_{22}$	<b>0,86</b>	0,90	0,70
$p_{35}r_{12}$	0,50	0,70	<b>0,65</b>

Nota: Los valores marcados son obtenidos mediante la imputación y/o asignación.

Los valores de los criterios obtenidos mediante la imputación o asignación, serán utilizados para calcular la prioridad o preferencia de procesos para cada nodo, multiplicando escalarmente cada vector de valoraciones de cada requerimiento 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.

#### 4.3.5. Cálculo de las prioridades o preferencias de los procesos para acceder a los recursos compartidos disponibles

Una vez calculada la prioridad según cada criterio y la prioridad nodal otorgada a cada requerimiento se procede a 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.

Posteriormente se calcula 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.

Las prioridades nodales tomadas fila por fila, es decir, respecto de cada recurso, se multiplicarán escalarmente por el vector de pesos finales normalizados, 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.

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 empate, podría resolverse de acuerdo al tiempo de creación de cada proceso, su prioridad inicial, u otra característica, en este caso se decide resolverlo por el método de FIFO (First in, First out - Primero en Entrar, Primero en Salir)

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) (ver **Tabla 17**).

La suma de todos estos productos en relación con el mismo recurso indicará la prioridad que deberá asignarse a este recurso, en relación con los demás recursos que también deberán asignarse.

**Tabla 17. Prioridades globales finales para asignar los recursos, constituye la Función de Asignación para Sistemas Distribuidos (FASD)**

FASD	Prioridad Global Final Para Asignar el Recurso	Proceso al que se asigna el recurso
$r_{11}$	0,3500	$r_{11}$ al $p_{37}$
$r_{12}$	0,4717	$r_{12}$ al $p_{37}$
$r_{13}$	0,3290	$r_{13}$ al $p_{13}$
$r_{21}$	0,3251	$r_{21}$ al $p_{37}$
$r_{22}$	0,3455	$r_{22}$ al $p_{34}$
$r_{23}$	0,2496	$r_{23}$ al $p_{11}$
$r_{24}$	0,1864	$r_{24}$ al $p_{34}$
$r_{31}$	0,3726	$r_{31}$ al $p_{34}$
$r_{32}$	0,3033	$r_{32}$ al $p_{34}$
$r_{33}$	0,4690	$r_{33}$ al $p_{23}$

El orden final de asignación de los recursos y los procesos destinatarios de los mismos se obtiene ordenando la **Tabla 17** (FASD), lo cual se muestra a continuación (ver **Tabla 18**).

**Tabla 18. Orden o prioridad final de asignación de los recursos y proceso al cual se asigna cada recurso, constituye la Función de Asignación para Sistemas Distribuidos Ordenada (FASDO)**

FASDO	Prioridad Global Final Para Asignar el Recurso	Proceso al que se asigna el recurso
$r_{12}$	0,4717	$r_{12}$ al $p_{37}$
$r_{33}$	0,4690	$r_{33}$ al $p_{23}$
$r_{31}$	0,3726	$r_{31}$ al $p_{34}$
$r_{11}$	0,3500	$r_{11}$ al $p_{37}$
$r_{22}$	0,3455	$r_{22}$ al $p_{34}$
$r_{13}$	0,3290	$r_{13}$ al $p_{13}$
$r_{21}$	0,3251	$r_{21}$ al $p_{37}$
$r_{32}$	0,3033	$r_{32}$ al $p_{34}$
$r_{23}$	0,2496	$r_{23}$ al $p_{11}$
$r_{24}$	0,1864	$r_{24}$ al $p_{34}$

El siguiente paso es reiterar el procedimiento como lo establece [10], 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 finaliza una vez que 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.

#### 4.3.6. Evaluación

El método aplicado en el modelo decisión fue muy efectivo, la diferencia que existe entre los valores originales en [10] y los valores que fueron imputados fueron mínimas. En esta propuesta se consideran como parte de la información histórica para aplicar una nueva imputación/asignación, los resultados de las imputaciones/asignaciones en los diferentes ciclos de recolección.

Una de las limitaciones más importantes al aplicar K-Means en esta propuesta, es que se ha establecido que tiene que estar disponible mínimamente el 70% de información de control en un registro. Una solución a esta problemática y a la situación en la que falten la totalidad de los datos que debería suministrar un nodo en un ciclo de sondeo, es aplicar un nuevo mecanismo que utiliza medias ponderadas, en los cuales los valores de los registros más recientes tendrán mayor peso que los demás.

## 5. Conclusiones y Líneas Futuras

La solución presentada en esta investigación en el contexto de sistemas distribuidos cuando la información sobre las variables que indican el estado de carga de alguno o algunos de los nodos llegue incompleta al nodo central, fue la de agregar una capa de imputación/asignación a un modelo de decisión. El método de imputación permite reemplazar los valores faltantes con valores estimados utilizando diferentes algoritmos, para que el modelo de decisión pueda establecer correctamente y de forma ordenada las asignaciones de recursos disponibles a los diferentes procesos, respetando la exclusión mutua.

Como línea futura y considerando las limitaciones que presenta el método de imputación K-Means (al igual que todos los métodos de imputación) cuando se dispone de pocos registros con los cuales operar para imputar, se pretende incorporar al modelo de decisión un mecanismo de imputación con la media ponderada, para solucionar el problema cuando la falta de información de control sea mayor al 30%, y en el momento que solo se disponga de información histórica de uno a diez ciclos de recolección.

## Agradecimientos

Este trabajo ha sido apoyado por el Proyecto: "Modelos de decisión para la gestión de recursos y procesos en sistemas distribuidos considerando migración de procesos, imputación de datos y lógica difusa en los nuevos

operadores de agregación", código 20F005 de la Universidad Nacional del Nordeste (Argentina).

## Referencias

- [1] A. Tanenbaum, G. Guerrero, and Ó. Velasco, *Sistemas operativos distribuidos*. México: Prentice-Hall Hispanoamericana S.A., 1996.
- [2] A. S. Tanenbaum, *Sistemas Operativos Modernos Tercera Edición*. 2009.
- [3] A. S. Tanenbaum and M. Van Steen, *Sistemas Distribuidos - Principios y Paradigmas*, 2da. México: Pearson Educación S. A., 2008.
- [4] D. L. La Red Martínez, *Sistemas Operativos*. EUDENE - Argentina, 2004.
- [5] D. Agrawal and A. El Abbadi, "An Efficient and Fault-Tolerant Solution for Distributed Mutual Exclusion," *ACM Trans. Comput. Syst.*, vol. 9, no. 1, 1991, doi: 10.1145/103727.103728.
- [6] G. Ricart and A. K. Agrawala, "An optimal algorithm for mutual exclusion in computer networks," *Commun. ACM*, vol. 24, no. 1, 1981, doi: 10.1145/358527.358537.
- [7] G. Cao and M. Singhal, "A delay-optimal quorum-based mutual exclusion algorithm for distributed systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 12, no. 12, 2001, doi: 10.1109/71.970560.
- [8] S. Lodha and A. Kshemkalyani, "A fair distributed mutual exclusion algorithm," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 6, 2000, doi: 10.1109/71.862205.
- [9] W. Stallings, *Sistemas operativos Aspectos internos y principios de diseño*, vol. Quinta Edición. 2005.
- [10] D. L. La Red Martínez, "Aggregation Operator for Assignment of Resources in Distributed Systems," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 10, 2017, doi: 10.14569/ijacsa.2017.081053.
- [11] D. L. La Red Martínez, F. Agostini, and C. Primorac, "Modelo de Asignación de Recursos para la Enseñanza de los Procesos Distribuidos," *Prim. Congr. Latinoam. Ing. - CLADI*, 2017.
- [12] D. L. La Red Martínez, J. C. Acosta, and F. Agostini, "Assignment of resources in distributed systems," in *IMCIC 2018 - 9th International Multi-Conference on Complexity, Informatics and Cybernetics, Proceedings*, 2018, vol. 2.
- [13] F. Agostini and D. L. La Red Martínez, "Allocation of shared resources," *14th Iber. Conf. Inf. Syst. Technol. - Cist. 2019*, pp. 1–6, Jun. 2019.
- [14] F. Agostini, D. L. la Red Martínez, and J. C. Acosta, "Assignment of resources in distributed systems with strict consensus requirements," in *IMCIC 2019 - 10th International Multi-Conference on Complexity, Informatics and Cybernetics, Proceedings*, 2019, vol. 1.
- [15] F. Agostini, D. L. La Red Martínez, and J. T. Fornerón Martínez, "Nuevo Operador De Agregación Para Grupos De Procesos," *Conferências IADIS Ibero-Americanas WWW/Internet e Comput. Apl. 2019*, pp. 223–230, 2019,

doi: 10.33965/ciaca2019\_2019141028.

- [16] P. Jönsson and C. Wohlin, "An evaluation of k-nearest neighbour imputation using Iikert data," 2004. doi: 10.1109/METRIC.2004.1357895.
- [17] Q. Song, M. Shepperd, and M. Cartwright, "A short note on safest default missingness mechanism assumptions," *Empir. Softw. Eng.*, vol. 10, no. 2, 2005, doi: 10.1007/s10664-004-6193-8.
- [18] B. Mehala and K. Vivekanandan, "An Analysis on K-Means Algorithm as an Imputation Method to Deal with Missing Values," *Asian J. Inf. Technol.*, vol. 7, pp. 434–441, 2008.
- [19] B. M. Patil, R. C. Joshi, and D. Toshniwal, "Missing value imputation based on k-mean clustering with weighted distance," *Commun. Comput. Inf. Sci.*, vol. 94 CCIS, no. PART 1, pp. 600–609, 2010, doi: 10.1007/978-3-642-14834-7\_56.
- [20] O. Vallejos, M. Valesani, and E. Rigonatto, "Técnicas de minería de datos aplicada a bases de datos imputadas," *Esc. Super. Politécnica del Litoral*, Apr. 2011, Accessed: Oct. 22, 2020. [Online]. Available: <http://www.revistas.espol.edu.ec/index.php/matematica/article/view/525>
- [21] M. M. Rahman and D. N. Davis, "Machine learning-based missing value imputation method for clinical datasets," *Lect. Notes Electr. Eng.*, vol. 229 LNEE, pp. 245–257, 2013, doi: 10.1007/978-94-007-6190-2-19.
- [22] F. Husson, J. Josse, B. Narasimhan, and G. Robin, "Imputation of Mixed Data With Multilevel Singular Value Decomposition," *J. Comput. Graph. Stat.*, vol. 28, no. 3, 2019, doi: 10.1080/10618600.2019.1585261.
- [23] S. J. Choudhury and N. R. Pal, "Imputation of missing data with neural networks for classification," *Knowledge-Based Syst.*, vol. 182, 2019, doi: 10.1016/j.knosys.2019.07.009.
- [24] C. R. Primorac, D. L. La Red Martínez, and M. E. Giovannini, "Metodología De Evaluación Del Desempeño De Métodos De Imputación Mediante Una Métrica Tradicional Complementada Con Un Nuevo Indicador," *Eur. Sci. J. ESJ*, vol. 16, no. 18, 2020, doi: 10.19044/esj.2020.v16n18p61.
- [25] J. Peláez, J. Dona, and D. La Red, "Majority opinion in group decision making using the qma-owa operator," pp. 449–454, 2004.
- [26] J. Peláez, J. M. Doña, and J. A. Gómez-Ruiz, "Analysis of OWA operators in decision making for modelling the majority concept," *Appl. Math. Comput.*, vol. 186, no. 2, 2007, doi: 10.1016/j.amc.2006.07.161.
- [27] Y. Dong, H. Zhang, and E. Herrera-Viedma, "Consensus reaching model in the complex and dynamic MAGDM problem," *Knowledge-Based Syst.*, vol. 106, 2016, doi: 10.1016/j.knosys.2016.05.046.
- [28] R. M. Schouten and G. Vink, "The Dance of the Mechanisms: How Observed Information Influences the Validity of Missingness Assumptions," *Sociol. Methods Res.*, 2018, doi: 10.1177/0049124118799376.
- [29] R. M. Schouten, P. Lugtig, and G. Vink, "Generating missing values for simulation purposes: a multivariate amputation procedure," *J. Stat. Comput. Simul.*, vol. 88, no. 15, pp. 2909–2930, 2018, doi: 10.1080/00949655.2018.1491577.
- [30] M. Leyva, N. González, J. Hechavarría, Y. Rivero, and J. Daher, "El diagnóstico de enfermedades desde el Análisis Inteligente de los Datos," *Rev. Espac.*, vol. 39 (Nº 28), p. 16, 2018.
- [31] D. B. Rubin, "Inference and missing data," *Biometrika*, vol. 63, no. 3. pp. 581–592, 1976. doi: 10.1093/biomet/63.3.581.
- [32] "The R Project for Statistical Computing." 2020.
- [33] Multivariate Imputation by Chained Equations, "No Title." 2020. [Online]. Available: <https://cran.r-project.org/web/packages/mice/index.html>
- [34] The R Project for Statistical Computing, "The R Project for Statistical Computing." 2020. [Online]. Available: <https://www.r-project.org/about.html>