

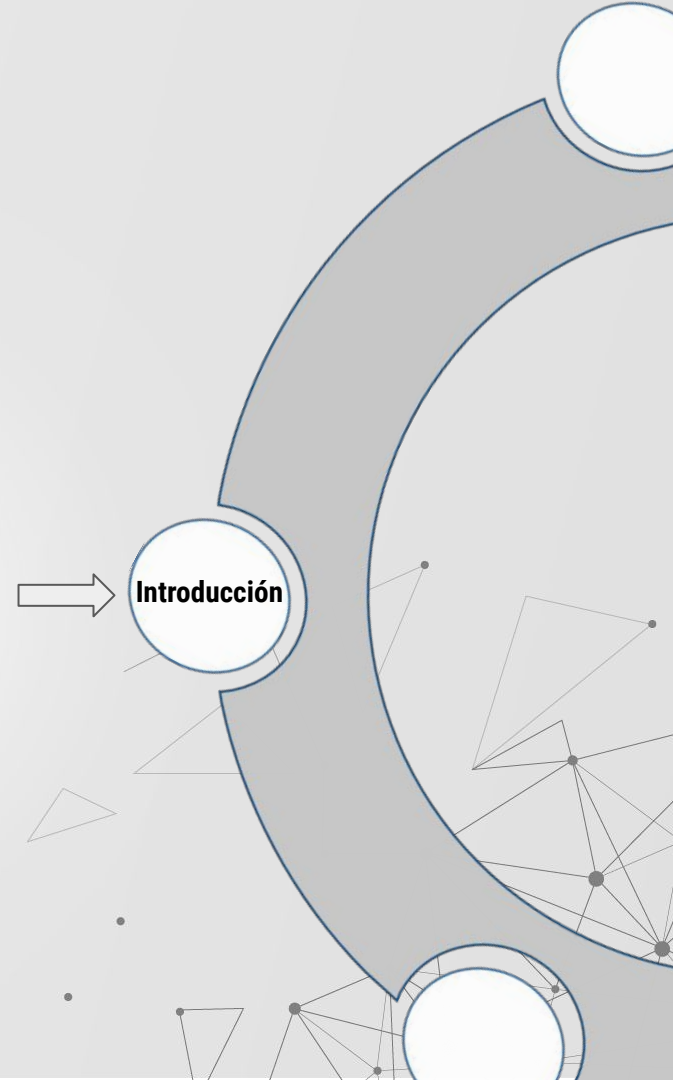
The background is a light gray with abstract geometric patterns. On the left, there is a dense network of dark gray lines connecting various sized black dots, forming a complex web. Scattered across the right side are several thin, light gray outlines of triangles of different sizes and orientations. In the top right corner, there are small, faint circles and dots.

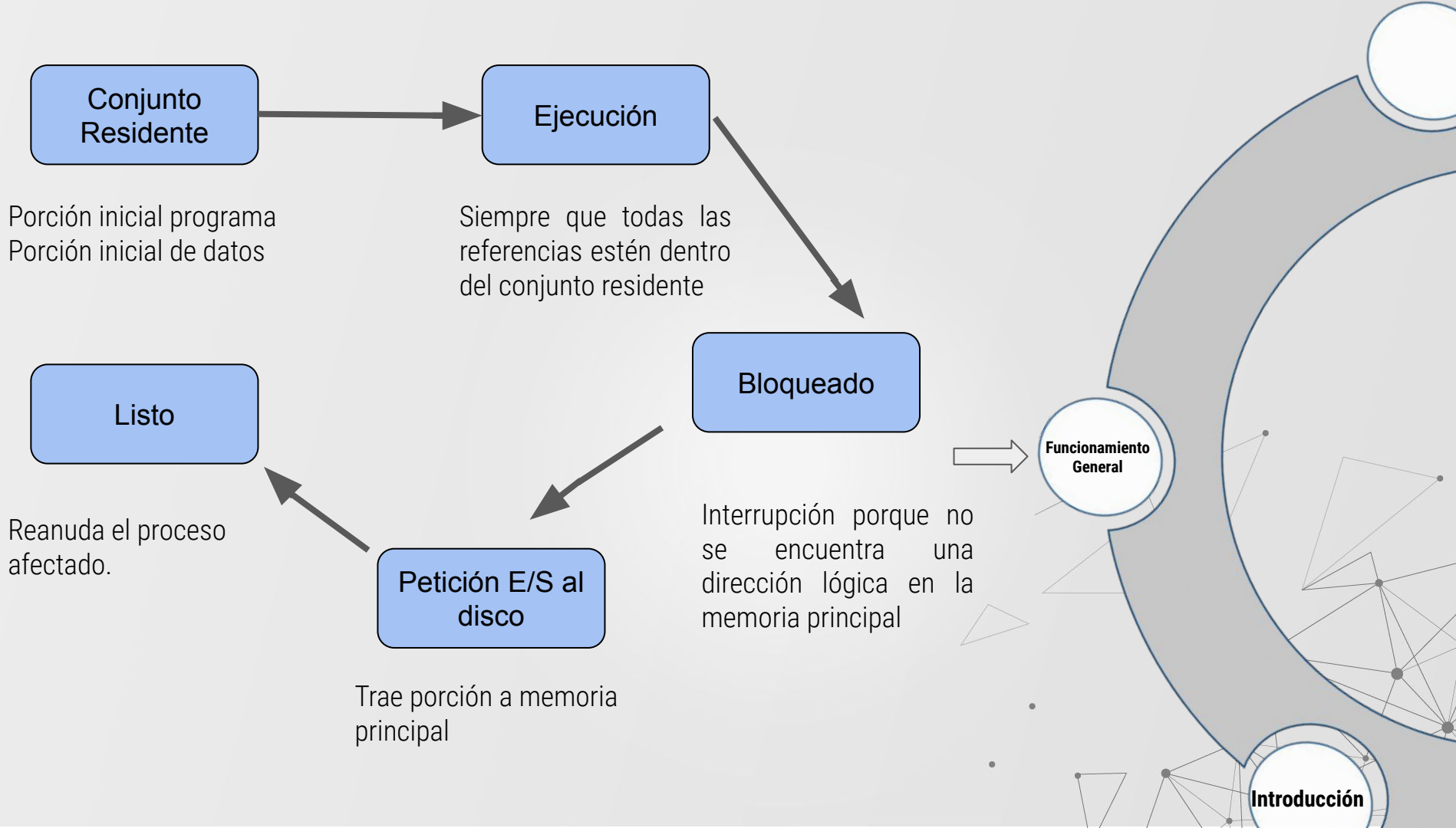
# Memoria Virtual

Cerrutti, Angie Lucía  
Folco, Juan Ignacio  
Menel, Angelo  
Vietto, Santiago

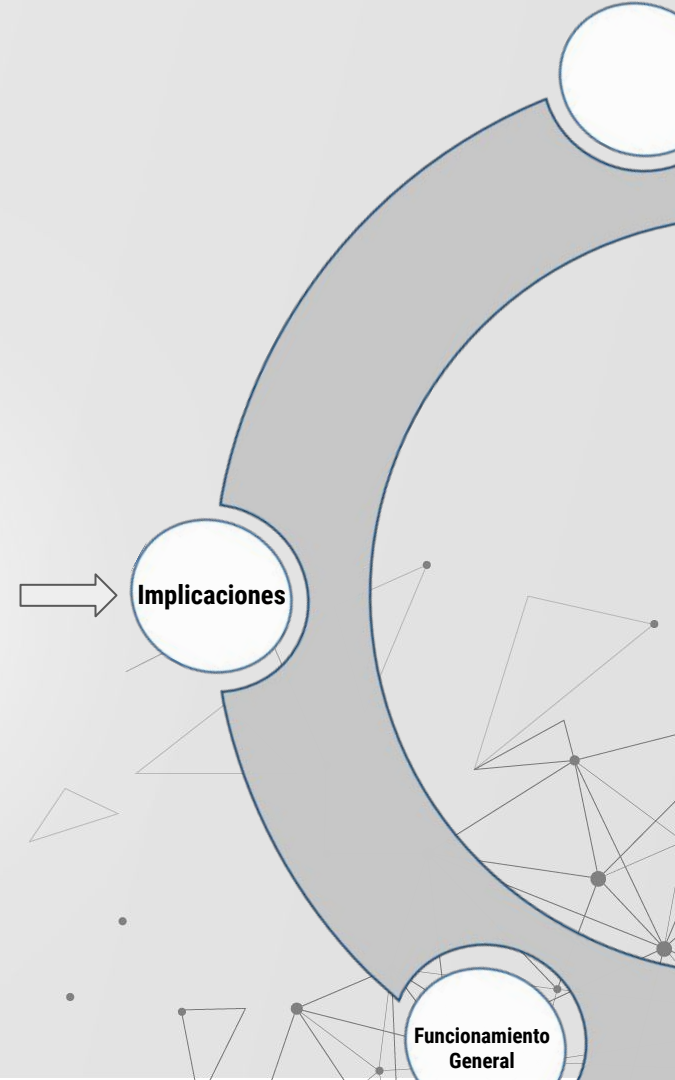
Es una técnica de gestión de la memoria que se encarga de que el sistema operativo disponga, tanto para el software de usuario como para sí mismo, de mayor cantidad de memoria que esté disponible físicamente.

Muchas aplicaciones requieren acceso a más información (código y datos) que la que se puede mantener en memoria física. Esto es así sobre todo cuando el sistema operativo permite múltiples procesos y aplicaciones ejecutándose simultáneamente. Una solución al problema de necesitar mayor cantidad de memoria de la que se posee consiste en que las aplicaciones mantengan parte de su información en disco, moviéndola a la memoria principal cuando sea necesario.



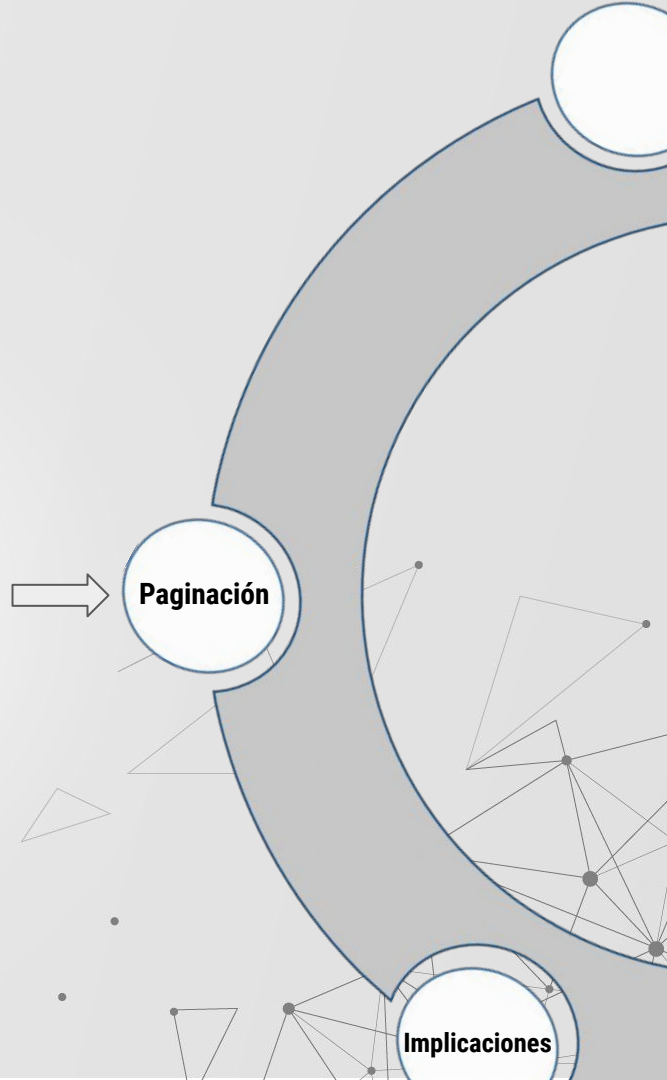


1. Pueden mantenerse un mayor número de procesos en memoria principal
  - Se cargan algunas porciones de los procesos a ejecutar.
2. Un proceso puede ser mayor que toda la memoria principal
  - Se delega el trabajo de fragmentar el proceso al sistema operativo y al hardware.



Proceso → Tabla de Páginas

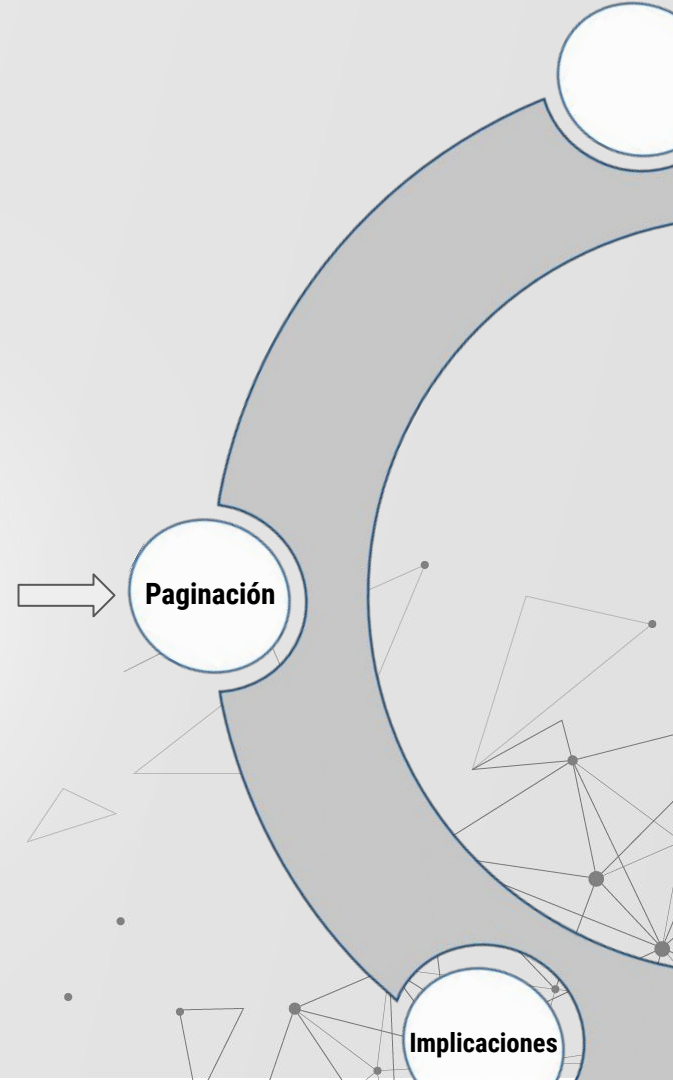
Sin embargo, las entradas de la tabla de páginas son más complejas:



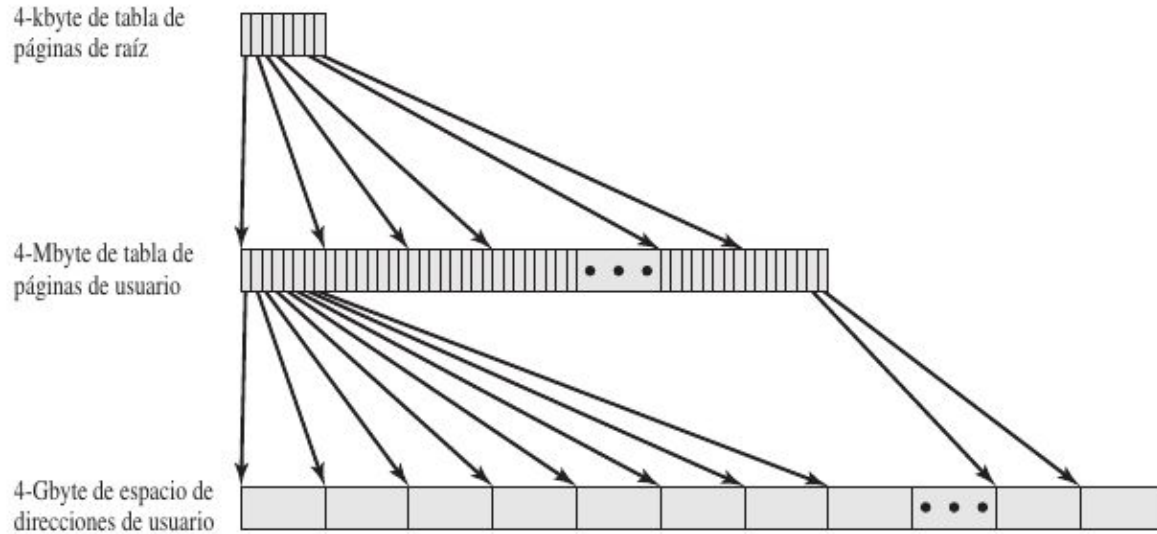
# Estructura de la tabla de páginas

El mecanismo básico de lectura de una palabra de la memoria implica la traducción de la dirección virtual, o lógica, consistente en un número de página y un desplazamiento, a la dirección física, consistente en un número de marco y un desplazamiento, usando para ello la tabla de páginas.

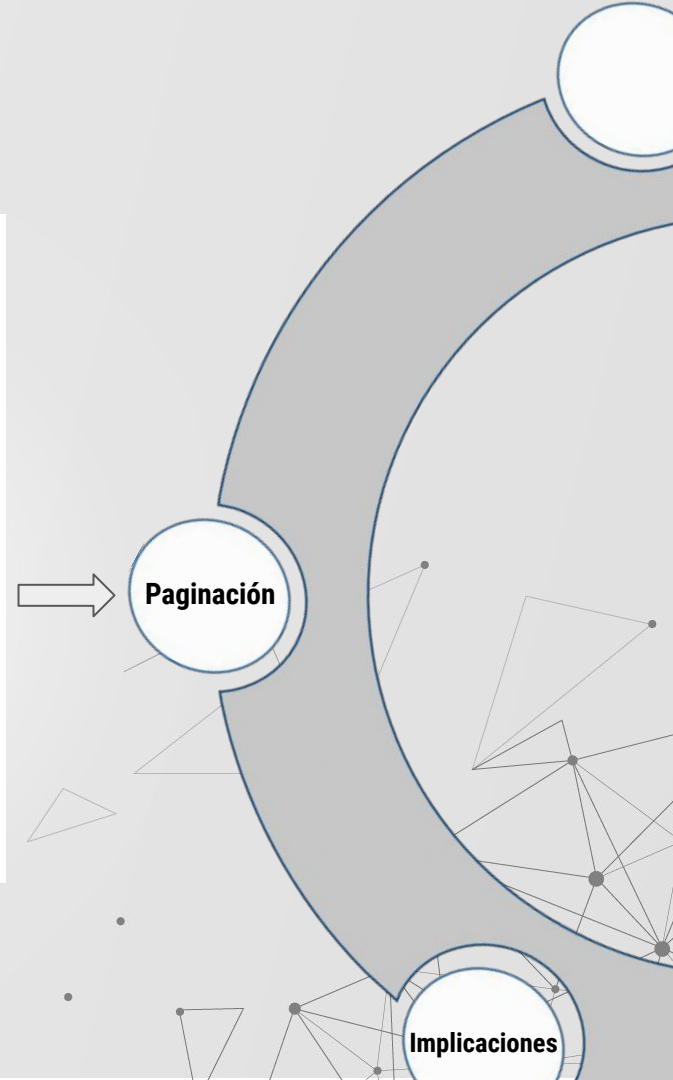
La mayoría de esquemas de memoria virtual almacena las tablas de páginas también en la memoria virtual, en lugar de en la memoria real.



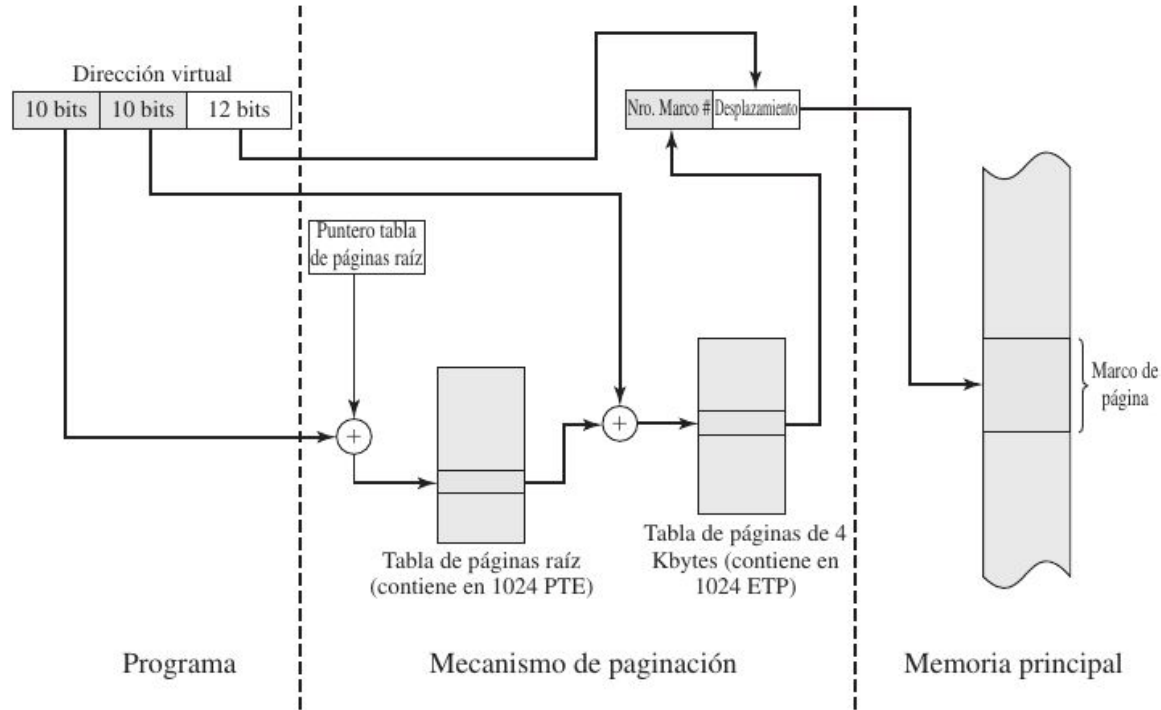
# Estructura de la tabla de páginas



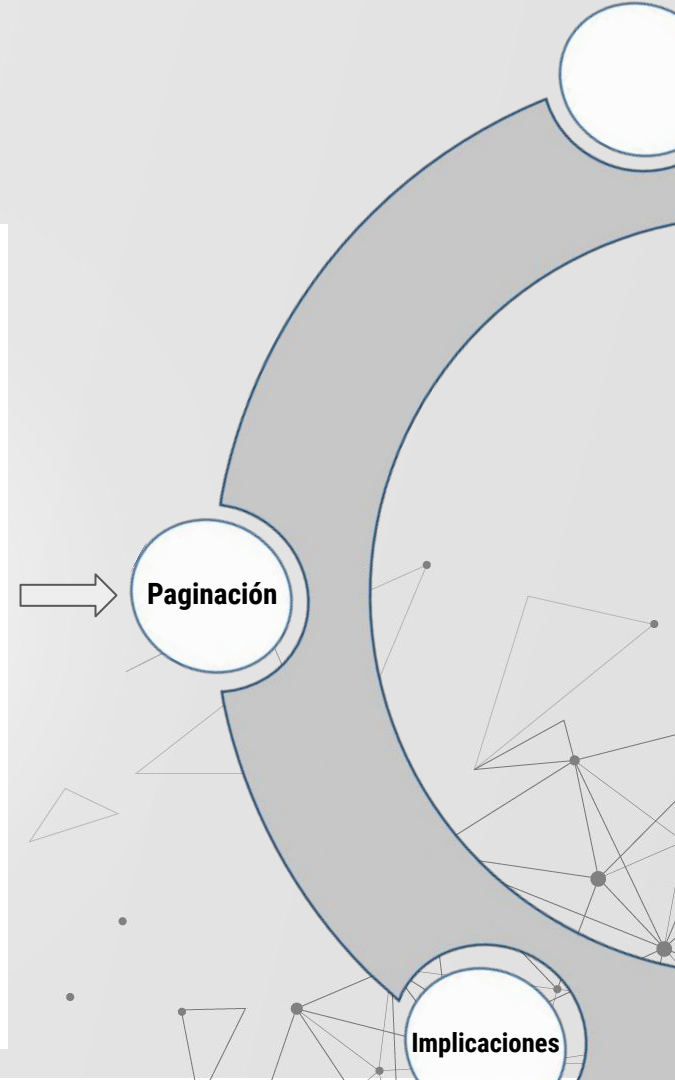
**Figura 8.4.** Una tabla de páginas jerárquica de dos niveles.



# Estructura de la tabla de páginas



**Figura 8.5.** Traducción de direcciones en un sistema de paginación de dos niveles.





# Tabla de páginas invertida

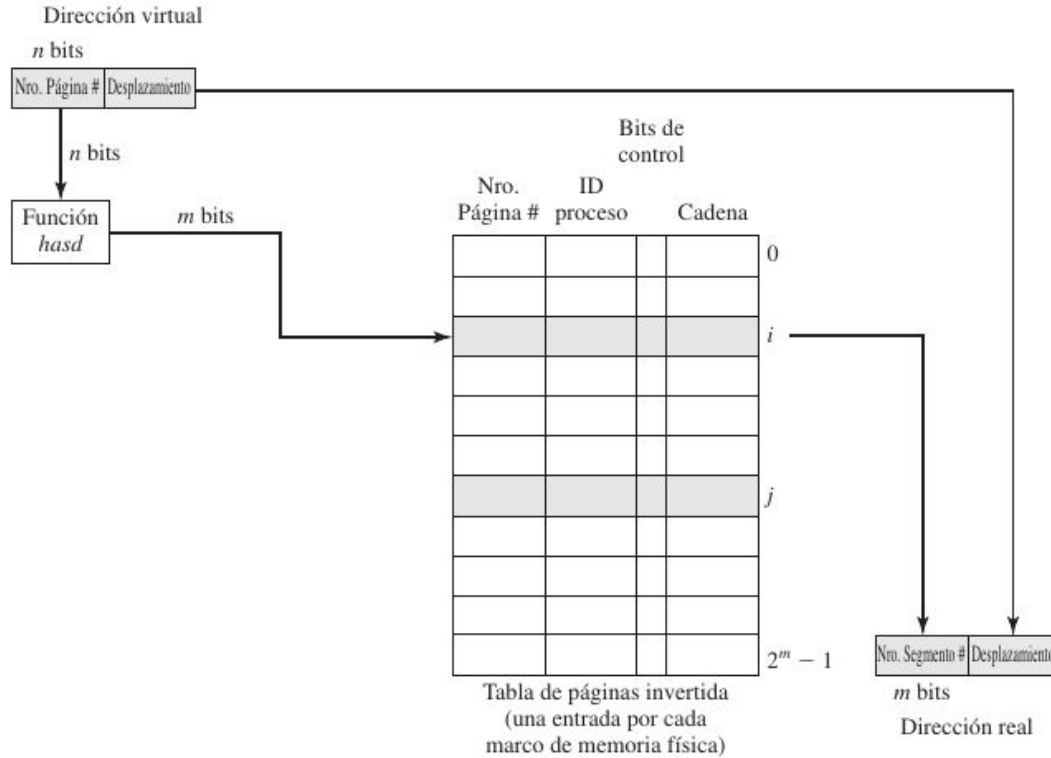
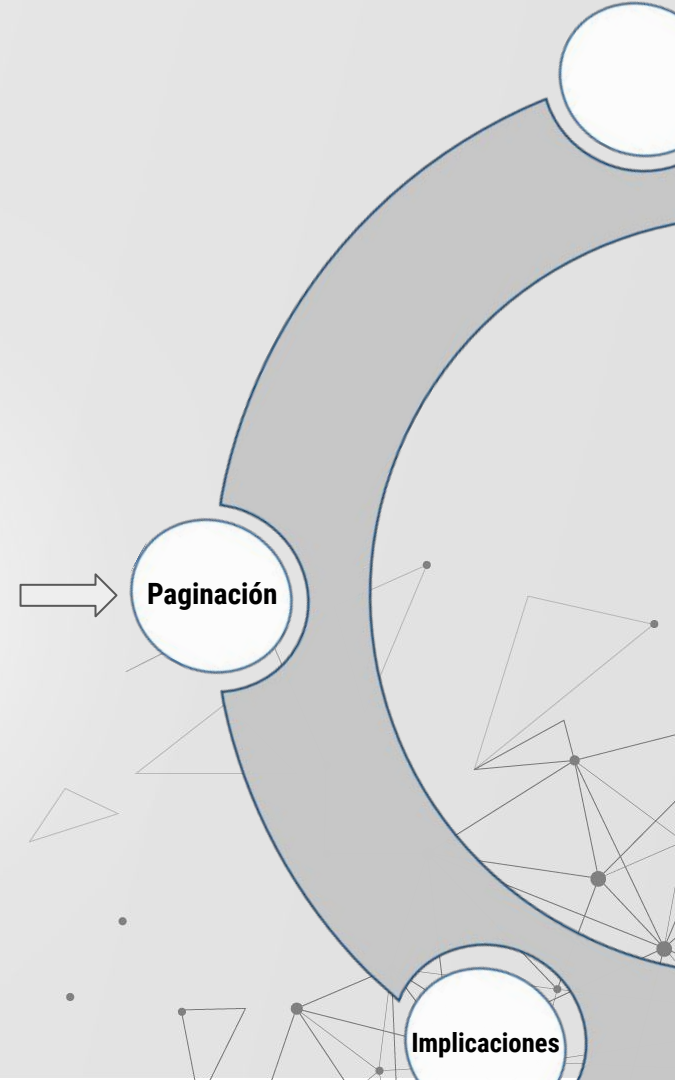


Figura 8.6. Estructura de tabla de páginas invertida.

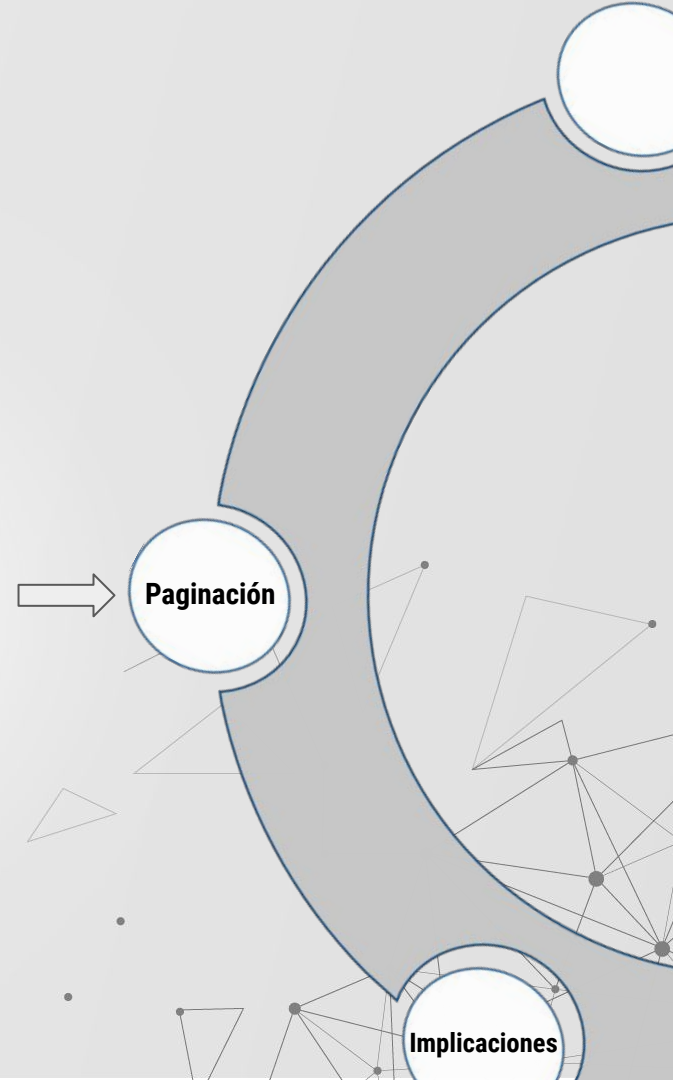


# Tabla de páginas invertida

Hay una entrada en la tabla de páginas invertida por cada marco de página real en lugar de uno por cada página virtual. De esta forma, lo único que se requiere para estas tablas de página siempre es una proporción fija de la memoria real, independientemente del número de procesos o de las páginas virtuales soportadas.

Debido a que más de una dirección virtual puede traducirse en la misma entrada de la tabla hash, una técnica de encadenamiento se utiliza para gestionar el desbordamiento.

Se indexan sus entradas de la tabla de páginas por el número de marco en lugar de por el número de página virtual.

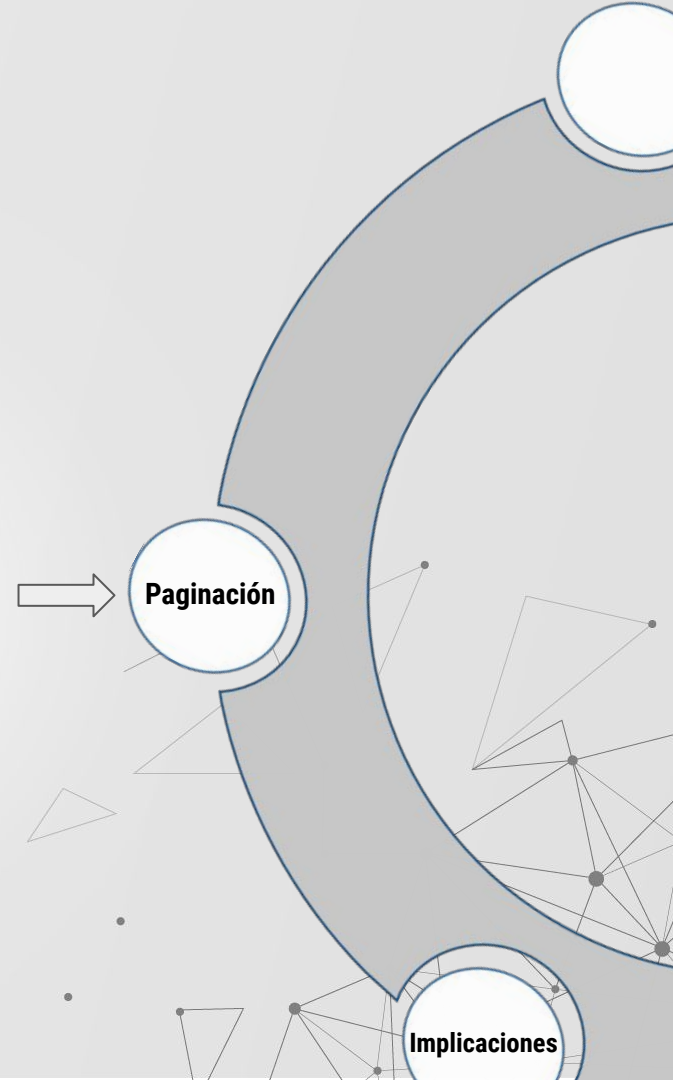


# Buffer de Traducción anticipada

Caché especial de alta velocidad.

Dos accesos a memoria física: uno para buscar la entrada en la tabla de páginas apropiada y otro para buscar los datos solicitados. duplicar el tiempo de acceso

Contiene aquellas entradas de la tabla de páginas que han sido usadas de forma más reciente.



# Buffer de Traducción anticipada

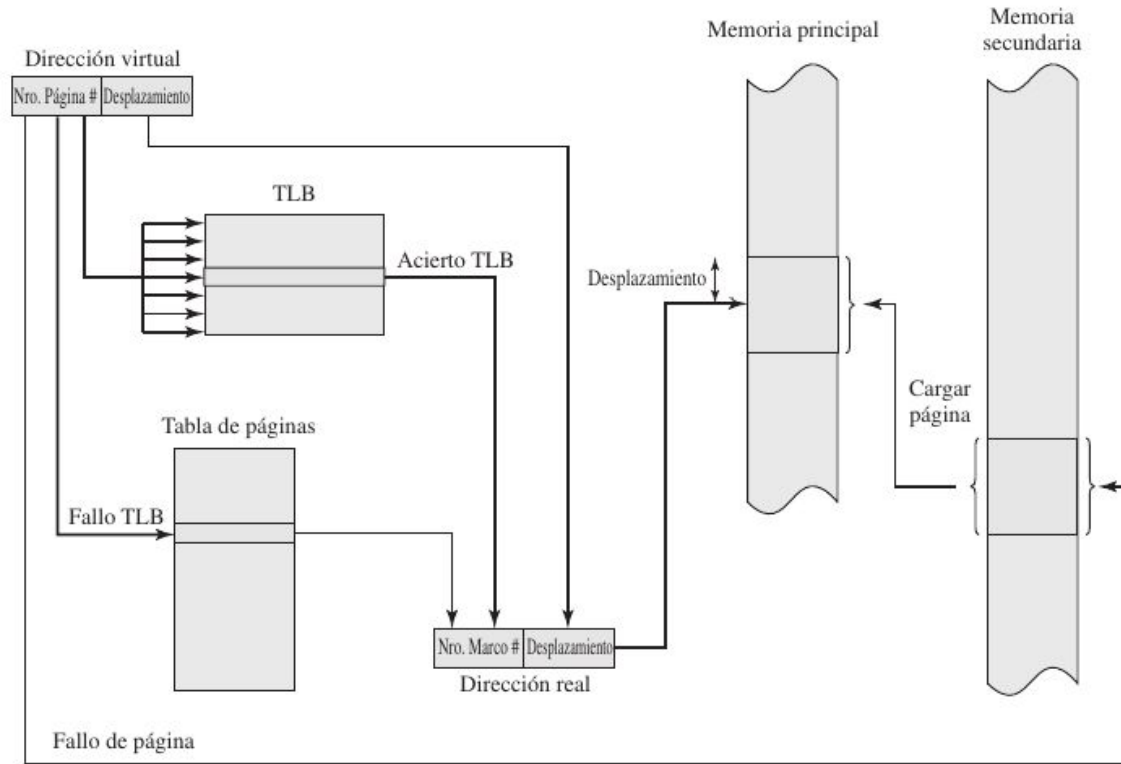
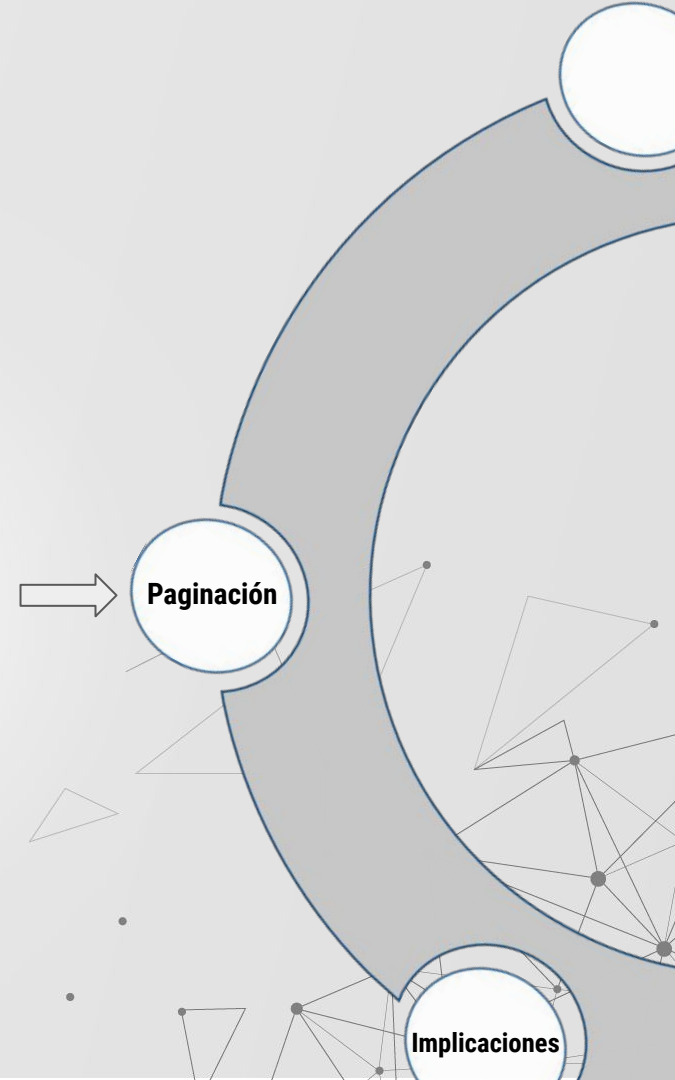


Figura 8.7. Uso de la TLB.



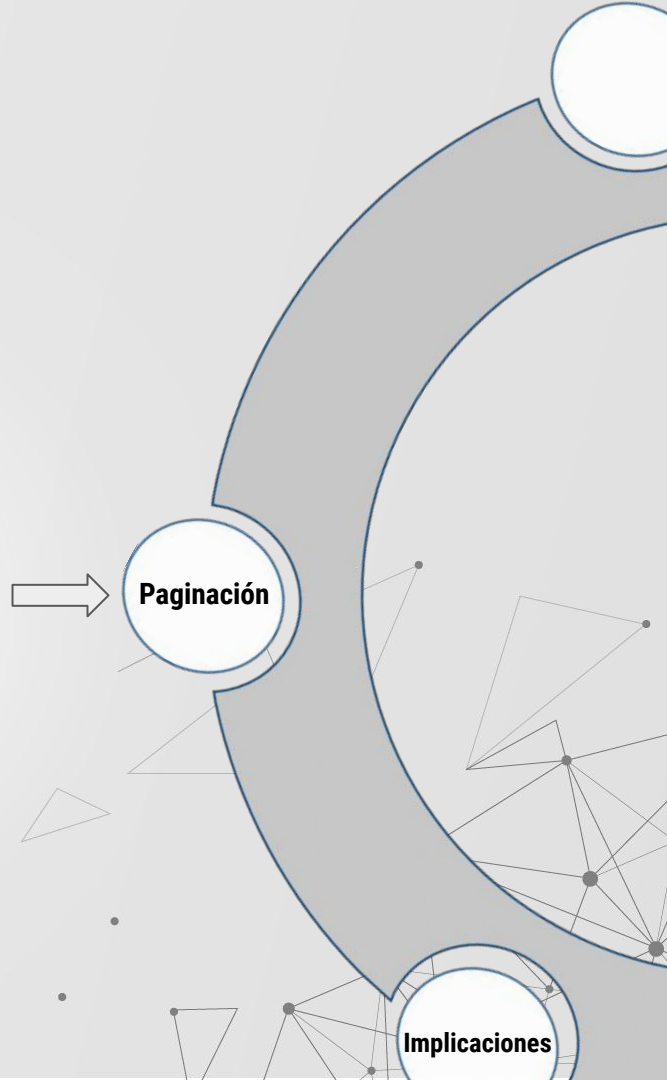
# Tamaño de página

Hay varios factores a considerar:

- Fragmentación interna:
  - Cuanto mayor es el tamaño de la página, menor cantidad de fragmentación interna.
  - Cuanto menor es la página, mayor número de páginas son necesarias para cada proceso.
- Características físicas de los dispositivos, permitiendo grandes tamaños de página para mejorar la eficiencia de transferencia de bloques de datos.

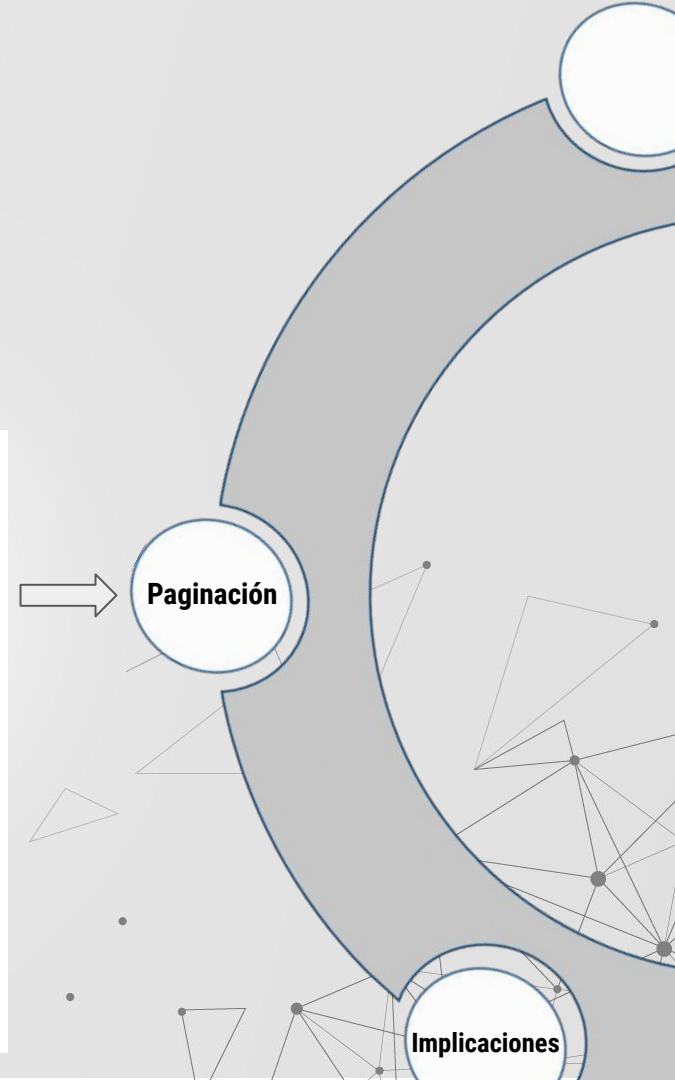
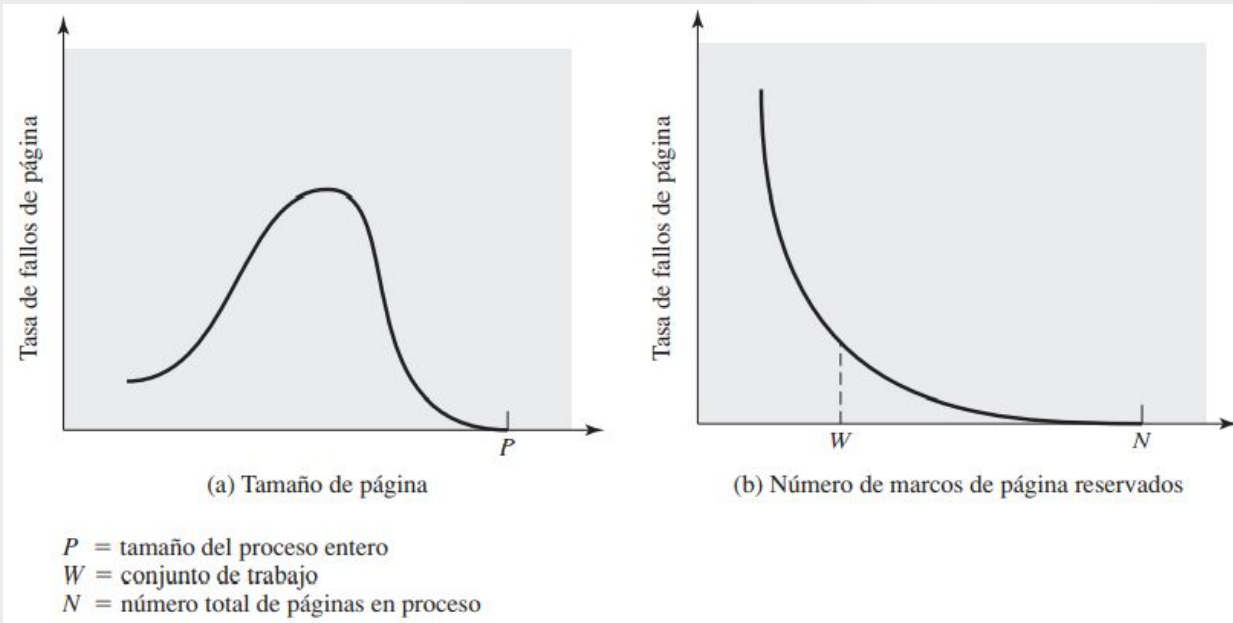
Si el tamaño de página es muy pequeño habrá un número relativamente alto de páginas disponibles en la memoria principal para cada proceso. Y después de un tiempo, cada página contendrá las partes de los procesos a las que se hizo referencia de forma reciente, logrando reducir la tasa de fallos de página.

A medida que el tamaño de páginas incrementa, la página misma contendrá información más lejana de la última referencia realizada (proceso), provocando que la tasa de fallos de página comience a crecer.



# Tamaño de página

- La tasa de fallos de página comenzará a caer a medida que el tamaño de la página se aproxima al tamaño del proceso completo.
- Para un tamaño de página fijo, la tasa de fallos cae a medida que el número de páginas mantenidas en la memoria principal crece.

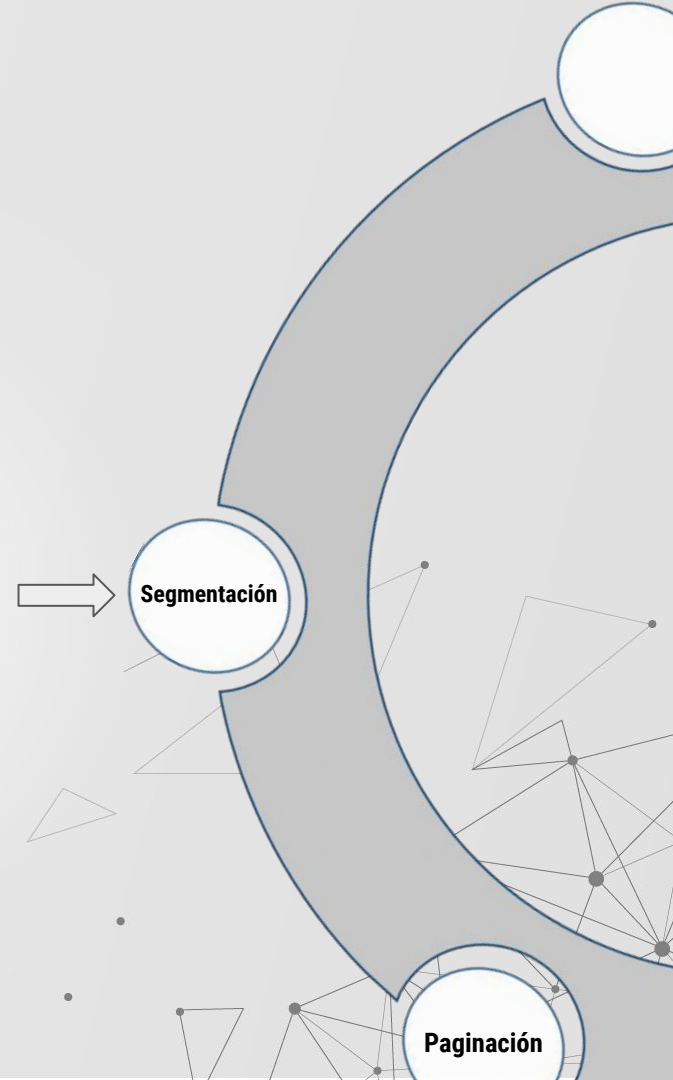


# Segmentación

Permite al programador ver la memoria como diferentes espacios de direcciones o segmentos, de diferentes tamaños y dinámicos. Se hace referencia a la memoria mediante un número de segmento, o desplazamiento.

Algunas ventajas:

1. Simplifica el tratamiento de estructuras de datos que pueden crecer, ya que al segmentar la memoria virtual, a una estructura de datos se le puede asignar su propio segmento, y el sistema operativo va a expandir o reducir el mismo según la necesidad.
2. Permite programas que se modifican o recompilan de forma independiente.
3. Da soporte a la compartición entre procesos. Por ejemplo, se puede situar un programa de uso común, en un segmento al que pueda hacerse referencia desde otros procesos.
4. Soporta mecanismos de protección, ya que sobre un segmento que puede definirse para contener un conjunto de programas, el programador puede asignar privilegios de acceso.



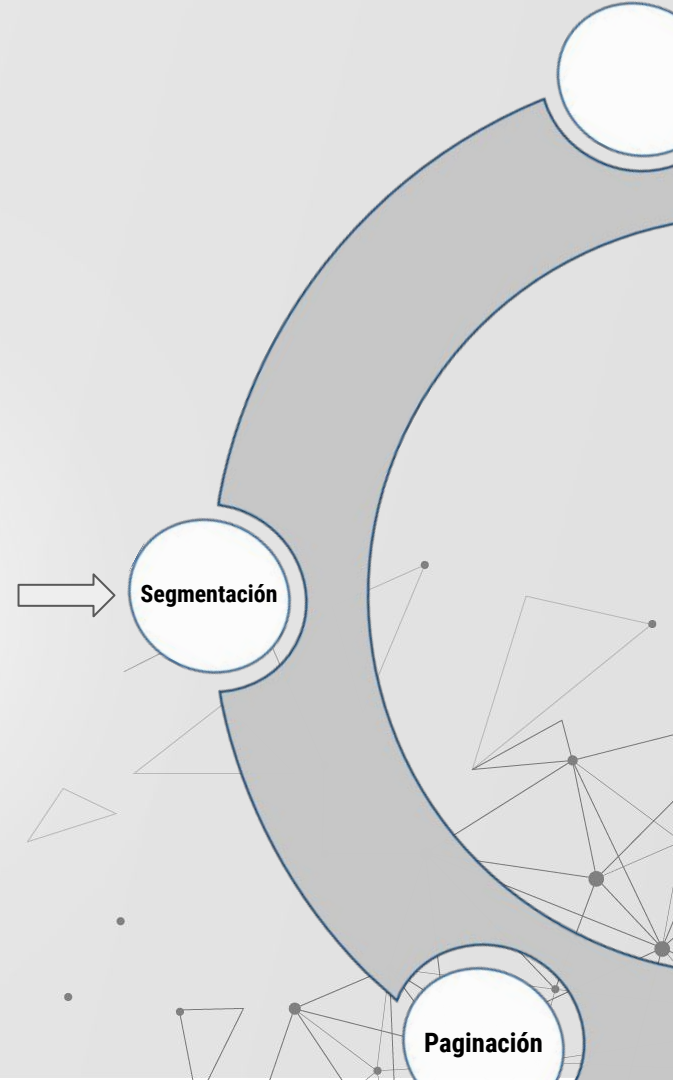
# Segmentación

## Organización:

- Cada proceso tiene su propia tabla de segmentos.
- Cuando todos los segmentos se han cargado en la memoria principal, la tabla de segmentos del proceso se crea y también se carga en dicha memoria.
- Cada entrada de la tabla de segmentos contiene la dirección de comienzo del correspondiente segmento en la memoria principal.
- Lo habitual es que haya una única tabla de segmentos por cada uno de los procesos.

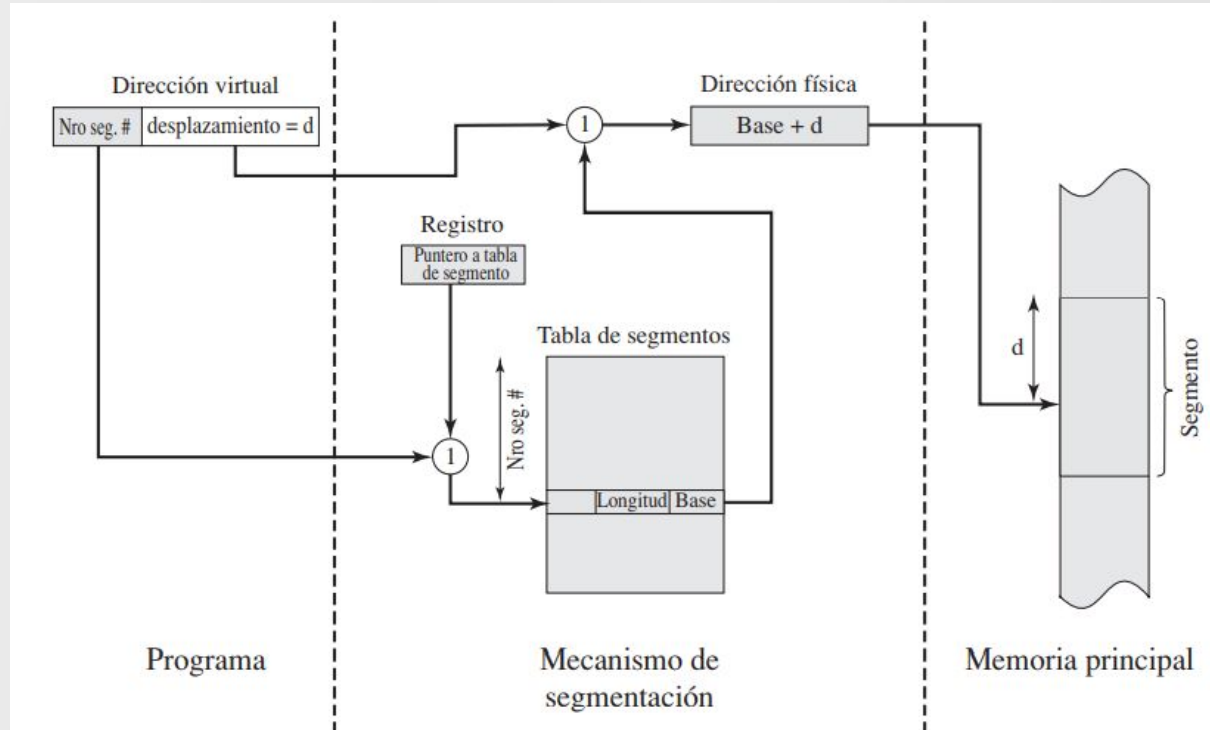
Traducción de direcciones en un sistema con segmentación: el mecanismo básico para la lectura de una palabra de memoria implica la traducción de una dirección virtual, consistente en un número de segmento y un desplazamiento, en una dirección física, usando la tabla de segmentos.

Como la tabla es de tamaño variable, dependiendo del tamaño del proceso, esta se debe encontrar en la memoria principal para poder acceder y no puede estar almacenada en un registro.





- Cuando un proceso está en ejecución, un registro mantiene la dirección de comienzo de la tabla de segmentos para dicho proceso.
- El número de segmento, de la dirección virtual, se utiliza para indexar esta tabla y para buscar la dirección de la memoria principal donde comienza dicho segmento.
- La tabla es añadida a la parte de desplazamiento de la dirección virtual, para producir la dirección real solicitada.



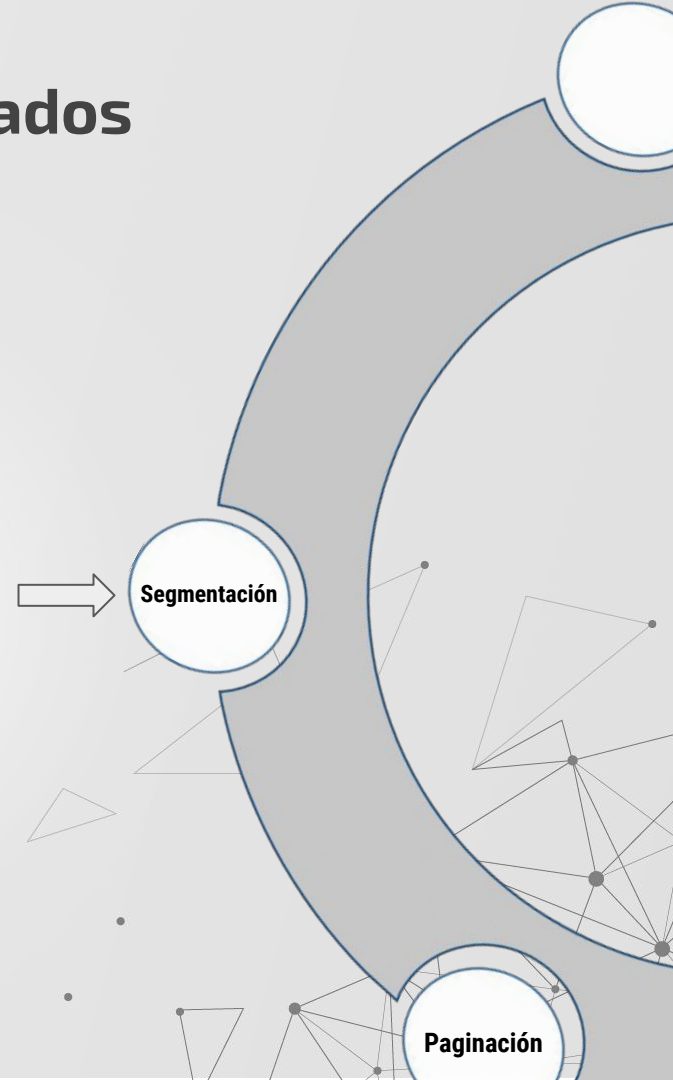
# Paginación y segmentación combinados

La paginación no es visible al programador y elimina la fragmentación externa, proporcionando un uso eficiente de la memoria principal.

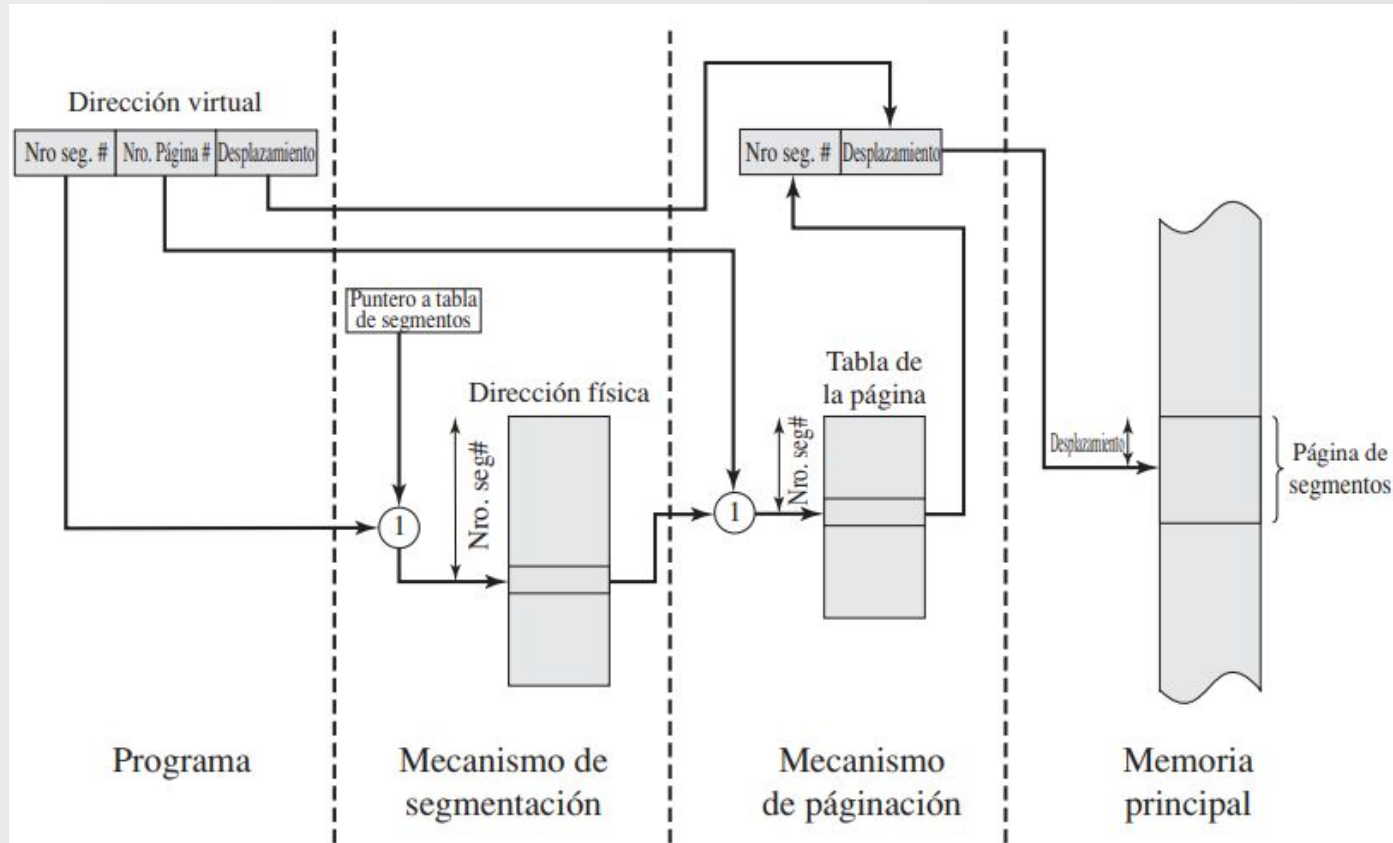
La segmentación sí es visible al programador, en donde además de sus beneficios, incluye la posibilidad de manejar estructuras de datos que crecen, modularidad, y dar soporte a la compartición y a la protección.

Traducción de direcciones en un sistema con segmentación/paginación:  
cada proceso tiene asociada una tabla de segmentos y varias tablas de páginas, una por cada uno de los segmentos.

- Cuando un proceso está en ejecución, un registro mantiene la dirección de comienzo de la tabla de segmentos de dicho proceso.
- A partir de la dirección virtual, el procesador utiliza la parte correspondiente al número de segmento para indexar dentro de la tabla de segmentos del proceso y luego encontrar la tabla de páginas de dicho segmento.



- Después, la parte correspondiente al número de página de la dirección virtual se utiliza para indexar la tabla de páginas y buscar el correspondiente número de marco que se combina con el desplazamiento correspondiente de la dirección virtual para generar la dirección real requerida.



The background features abstract geometric patterns. On the left, there are two large, complex wireframe structures resembling interconnected triangles or a mesh. On the right, there are several smaller, isolated triangles and a few scattered dots, suggesting a sparse network or data points.

# **Software del sistema operativo**

# Software del sistema operativo

El diseño de la parte de la gestión de la memoria del sistema operativo depende de tres opciones fundamentales a elegir:

- Si el sistema usa o no técnicas de memoria virtual.
- El uso de paginación o segmentación o ambas.
- **Los algoritmos utilizados para los diferentes aspectos de la gestión de la memoria.**

Las elecciones posibles para las dos primeras opciones dependen de la plataforma de hardware disponible.

Las elecciones relativas a la tercera opción entran dentro del dominio del software del sistema operativo.



# Políticas

**Tabla 8.3.** Políticas del sistema operativo para la memoria virtual.

<b>Política de recuperación</b> Bajo demanda Paginación adelantada	<b>Gestión del conjunto residente</b> Tamaño del conjunto residente Fijo Variable Ámbito de reemplazo Global Local
<b>Política de ubicación</b>	
<b>Política de reemplazo</b> Algoritmos básicos Óptimo FIFO Usada menos recientemente (LRU) Del reloj <i>Buffers</i> de página	<b>Política de limpieza</b> Bajo demanda Limpieza adelantada <b>Control de carga</b> Grado de multiprogramación

# Política de recuperación

- Determina cuándo una página se trae a la memoria principal. Las dos alternativas habituales son **bajo demanda** y **paginación adelantada (*prepaging*)**.

Con **paginación bajo demanda**, una página se trae a memoria sólo cuando se hace referencia a una posición en dicha página.

Con **paginación adelantada (*prepaging*)**, se traen a memoria también otras páginas, diferentes de la que ha causado el fallo de página. Si las páginas de un proceso se encuentran almacenadas en la memoria secundaria de forma contigua, es mucho más eficiente traer a la memoria un número de páginas contiguas de una vez, en lugar de traerlas una a una a lo largo de un periodo de tiempo más amplio.



# Política de ubicación

- En qué parte de la memoria real van a residir las porciones de la memoria de un proceso.

En los sistemas de segmentación puros, la política de ubicación es un aspecto de diseño muy importante; sin embargo, para sistemas que usan o bien paginación pura o paginación combinada con segmentación, la ubicación es *habitualmente irrelevante* debido a que el hardware de traducción de direcciones y el hardware de acceso a la memoria principal pueden realizar sus funciones en cualquier combinación de página-marco con la misma eficiencia.





# Política de reemplazo

- Entre el conjunto de páginas a considerar, qué página en concreto es la que se va a reemplazar.

Cuando todos los marcos de la memoria principal están ocupados y es necesario traer una nueva página para resolver un fallo de página, la política de reemplazo determina qué página de las que actualmente están en memoria va a reemplazarse.

Todas las políticas tienen como objetivo que la página que va a eliminarse sea aquella que tiene menos posibilidades de volver a tener una referencia en un futuro próximo.



# Política de reemplazo - Bloqueo de marcos (pre-algoritmo)

Restricción que se aplica a las políticas de reemplazo antes de indagar en los diferentes algoritmos: algunos marcos de la memoria principal pueden encontrarse bloqueados.

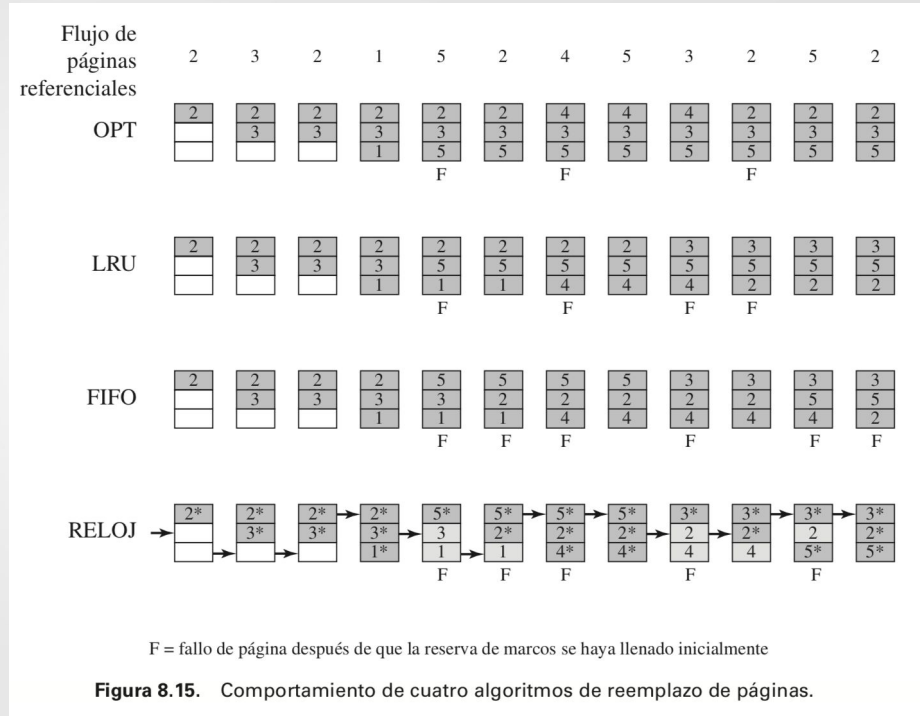
Cuando un marco está bloqueado, la página actualmente almacenada en dicho marco no puede reemplazarse.

El bloqueo se puede realizar asociando un bit de bloqueo a cada uno de los marcos. Este bit se puede almacenar en la tabla de marcos o también incluirse en la tabla de páginas actual.



# Política de reemplazo - Algoritmos básicos

- Óptimo.
- Usado menos recientemente (least recently used-LRU).
- FIFO (first-in-first-out).
- Reloj.



**Figura 8.15.** Comportamiento de cuatro algoritmos de reemplazo de páginas.

# Política de reemplazo - Algoritmos básicos - óptimo

Se toma como reemplazo la página para la cuál el instante de la siguiente referencia se encuentra más lejos.

Los resultados son el menor número de posibles fallos de página.

**Imposible de implementar.** porque requiere que el sistema operativo tenga un perfecto conocimiento de los eventos futuros.

Flujo de páginas referenciales	2	3	2	1	5	2	4	5	3	2	5	2																																				
OPT	<table><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table><tr><td>4</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	4	3	5	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5	<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>5</td></tr></table>	2	3	5
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
				F		F				F																																						

# Política de reemplazo - Algoritmos básicos - LRU

Se toma como reemplazo la página de memoria que no se haya referenciado desde hace más tiempo.

Proporciona unos resultados casi tan buenos como la política óptima.

Posee dificultad en su implementación, impone una sobrecarga significativa.

Flujo de páginas referenciales	2	3	2	1	5	2	4	5	3	2	5	2																																				
LRU	<table><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table> <div>F</div>	2	5	1	<table><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>1</td></tr></table>	2	5	1	<table><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> <div>F</div>	2	5	4	<table><tr><td>2</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table>	2	5	4	<table><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> <div>F</div>	3	5	4	<table><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table> <div>F</div>	3	5	2	<table><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2	<table><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table>	3	5	2
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
1																																																
2																																																
5																																																
4																																																
2																																																
5																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																
3																																																
5																																																
2																																																

# Política de reemplazo - Algoritmos básicos - FIFO

Trata los marcos de página ocupados como si se tratase de un buffer circular, y las páginas se reemplazan mediante una estrategia cíclica de tipo round-robin. Solo necesita un puntero que recorra de forma circular los marcos de página del proceso.

**Sencilla de implementar.** El razonamiento tras este modelo, además de su simplicidad, es el reemplazo de la página que lleva en memoria más tiempo.

Rendimiento relativamente pobre.

Flujo de páginas referenciales	2	3	2	1	5	2	4	5	3	2	5	2																																				
FIFO	<table><tr><td>2</td></tr><tr><td></td></tr><tr><td></td></tr></table>	2			<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td></td></tr></table>	2	3		<table><tr><td>2</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table>	2	3	1	<table><tr><td>5</td></tr><tr><td>3</td></tr><tr><td>1</td></tr></table> F	5	3	1	<table><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>1</td></tr></table> F	5	2	1	<table><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table> F	5	2	4	<table><tr><td>5</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table>	5	2	4	<table><tr><td>3</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table> F	3	2	4	<table><tr><td>3</td></tr><tr><td>2</td></tr><tr><td>4</td></tr></table>	3	2	4	<table><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>4</td></tr></table> F	3	5	4	<table><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>2</td></tr></table> F	3	5	2
2																																																
2																																																
3																																																
2																																																
3																																																
2																																																
3																																																
1																																																
5																																																
3																																																
1																																																
5																																																
2																																																
1																																																
5																																																
2																																																
4																																																
5																																																
2																																																
4																																																
3																																																
2																																																
4																																																
3																																																
2																																																
4																																																
3																																																
5																																																
4																																																
3																																																
5																																																
2																																																

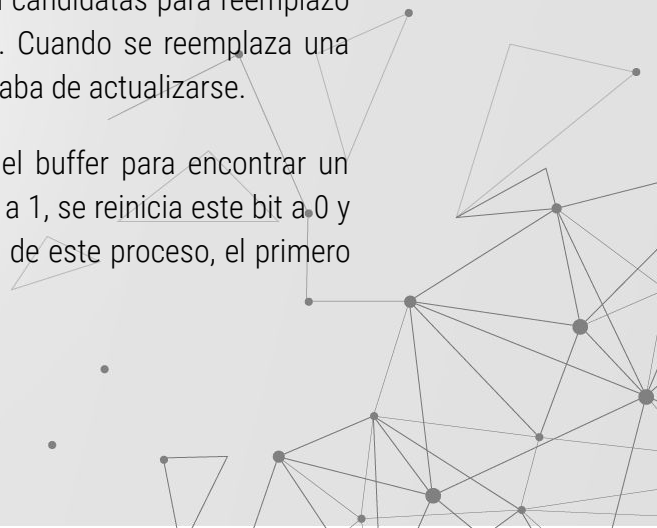
# Política de reemplazo - Algoritmos básicos - Reloj

Esquema originado de intentos en mejorar las políticas existentes.

En su forma más sencilla requiere la inclusión de un bit adicional en cada uno de los marcos de página, denominado bit de usado. Cuando una página se trae por primera vez a la memoria, el bit de usado de dicho marco se pone a 1. En cualquier momento que la página vuelva a utilizarse su bit de usado se pone a 1.

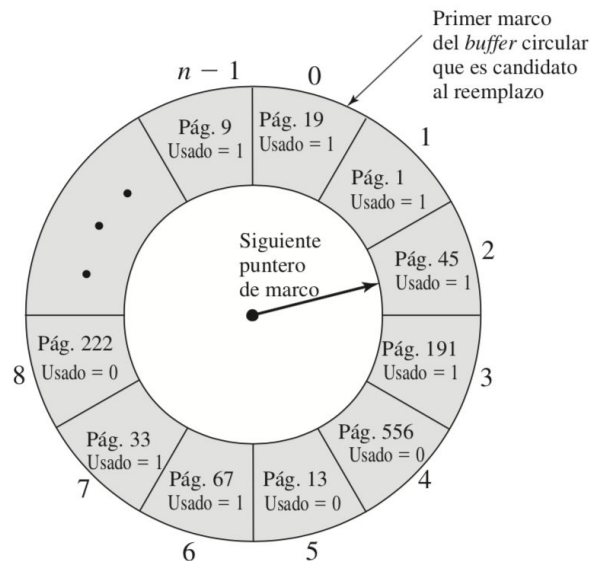
Para el algoritmo de reemplazo de páginas, el conjunto de todas las páginas que son candidatas para reemplazo se disponen como si se tratase de un buffer circular, al cual se asocia un puntero. Cuando se reemplaza una página, el puntero indica el siguiente marco del buffer justo después del marco que acaba de actualizarse.

Cuando llega el momento de reemplazar una página, el sistema operativo recorre el buffer para encontrar un marco con su bit de usado a 0. Cada vez que encuentra un marco con el bit de usado a 1, se reinicia este bit a 0 y se continúa. Si alguno de los marcos del buffer tiene el bit de usado a 0 al comienzo de este proceso, el primero de estos marcos que se encuentre se seleccionará para reemplazo.

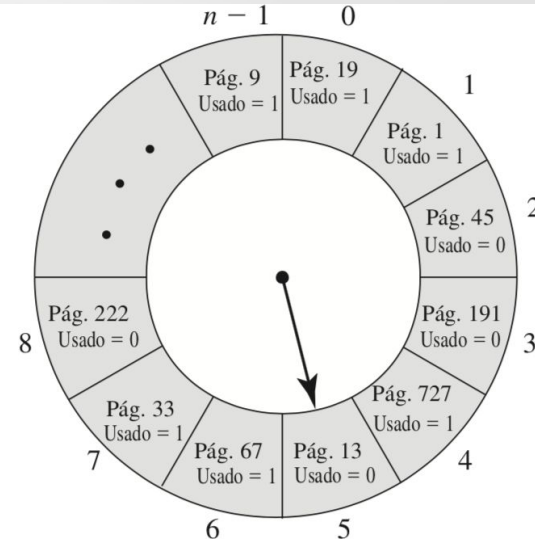


# Política de reemplazo - Algoritmos básicos - Reloj

Un buffer circular con  $n$  marcos de memoria principal que se encuentran disponibles para reemplazo de la página.



(a) Estado del buffer justo antes del reemplazo de una página



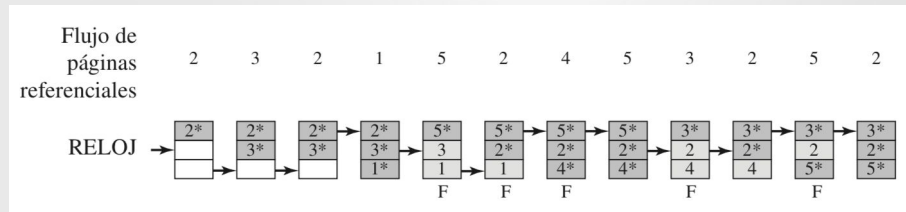
(b) Estado del buffer justo después del siguiente reemplazo de página

**Figura 8.16.** Ejemplo de operación de la política de reemplazo del reloj.



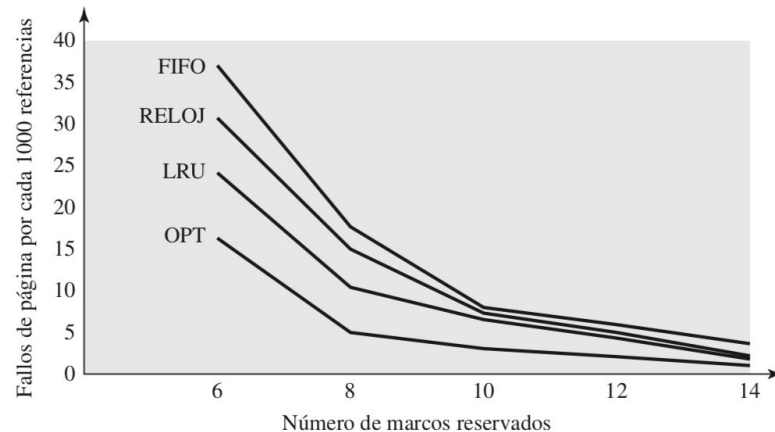
# Política de reemplazo - Algoritmos básicos - Reloj

Figura del comportamiento de un reloj. La presencia de un asterisco indica que el correspondiente bit de usado es igual a 1, y la flecha indica cuál es la posición actual del puntero. Nótese que la política del reloj intenta proteger los marcos 2 y 5 de un posible reemplazo.



# Política de reemplazo - Algoritmos básicos

La Figura muestra los resultados de un experimento que compara los cuatro algoritmos; se asume que el número de marcos de página asignados a cada proceso es fijo. El resultado se basa en la ejecución de  $0,25 \times 10^6$  referencias en un programa FORTRAN, utilizando un tamaño de página de 256 palabras. Las diferencias entre las cuatro políticas son más palpables cuando el número de marcos reservados es pequeño, estando FIFO por encima en un factor de 2 veces peor que el óptimo.



**Figura 8.17.** Comparativa de algoritmos de reemplazo local con reserva de marcos fija.

# Política de reemplazo - Buffering de páginas

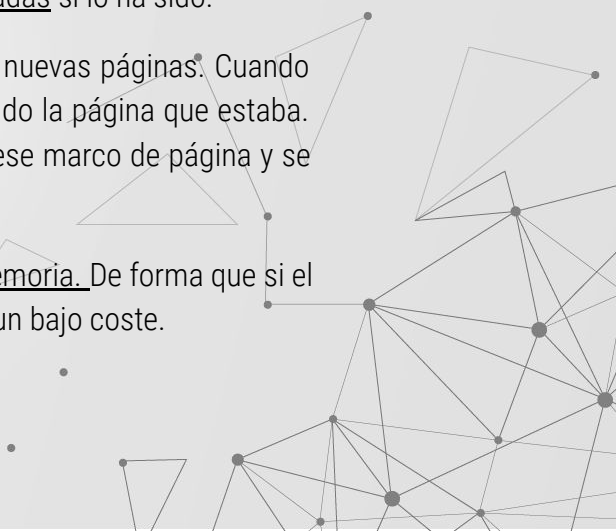
Es una estrategia que puede mejorar el rendimiento de la paginación y que permite el uso de una política de reemplazo de páginas sencilla es el buffering.

El algoritmo de reemplazo de páginas es el FIFO sencillo.

Para mejorar el rendimiento, una página remplazada no se pierde sino que se asigna a una de las dos siguientes listas: Lista de páginas libres si la página no se ha modificado o Lista de páginas modificadas si lo ha sido.

La lista de páginas libres es una lista de marcos de páginas disponibles para lectura de nuevas páginas. Cuando una página se va a leer, se utiliza el marco de página en la cabeza de esta lista, eliminando la página que estaba. Cuando se reemplaza una página que no se ha modificado, se mantiene en la memoria ese marco de página y se añade al final de la lista de páginas libres.

El aspecto más importante es que la página que se va a reemplazar se mantiene en la memoria. De forma que si el proceso hace referencia a esa página, se devuelve al conjunto residente del proceso con un bajo coste.



# Política de reemplazo - tamaño de la caché

En sistemas que utilizan algún tipo de buffering de páginas, se puede mejorar el rendimiento de la caché añadiendo a la política de reemplazo de páginas una política de ubicación de páginas en el buffer. La mayoría de sistemas operativos ubican las páginas seleccionando un marco procedente del buffer de páginas, de forma arbitraria; utilizando habitualmente una disciplina de tipo FIFO.

La esencia de esta estrategia consiste en traer páginas consecutivas a la memoria principal de forma que se minimice el número de marcos de página que se encuentran proyectados en las mismas ranuras de la caché.



# Gestión del conjunto residente

El sistema operativo debe decidir cuánta memoria asigna a un proceso en particular cuando dicho proceso se trae a la memoria.



# Gestión del conjunto residente - Tamaño del conjunto residente

Con la memoria virtual paginada, no es necesario traer todas las páginas de un proceso a la memoria principal para preparar su ejecución debido a que el sistema operativo debería saber cuántas páginas traerse, esto es, cuánta memoria principal debería reservar para un proceso en particular. Diferentes factores entran en juego:

- Cuanto menor es la cantidad de memoria reservada para un proceso, mayor es el número de procesos que pueden residir en la memoria principal a la vez.
- Si el conjunto de páginas de un proceso que están en memoria es relativamente pequeño, entonces, en virtud del principio de proximidad de referencia, la posibilidad de un fallo de página es mayor.



# Gestión del conjunto residente - Tamaño del conjunto residente

Teniendo en cuenta estos factores, se pueden encontrar dos tipos de políticas existentes en los sistemas operativos contemporáneos.

La política de asignación fija proporciona un número fijo de marcos de memoria principal disponibles para ejecución. Este número se decide en el momento de la carga inicial de proceso (instante de creación del proceso).

Una política de asignación variable permite que se reserven un número de marcos por proceso que puede variar a lo largo del tiempo de vida del mismo.



# Gestión del conjunto residente - Ámbito de reemplazo

La estrategia del ámbito de reemplazo se puede clasificar en global y local. Ambos tipos de políticas se activan por medio de un fallo de página cuando no existen marcos de página libres.

Una política de reemplazo local selecciona únicamente entre las páginas residentes del proceso que ha generado el fallo de página.

Para la identificación de la página a reemplazar en una política de reemplazo global se consideran todas las páginas en la memoria principal que no se encuentren bloqueadas como candidatos para el reemplazo, independientemente de a qué proceso pertenezca cada página en particular.

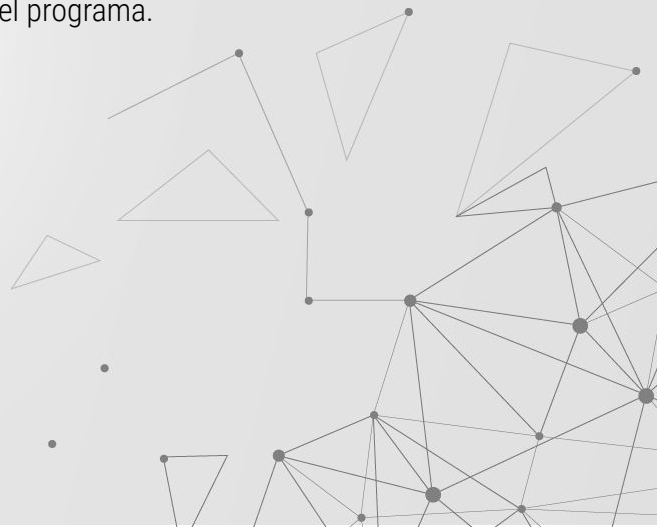




# Gestión del conjunto residente - Asignación fija, ámbito local

Se parte de un proceso que se encuentra en ejecución en la memoria principal con un número de marcos fijo. Cuando se da un fallo de página, el sistema operativo debe elegir una página entre las residentes del proceso actual para realizar el reemplazo. Se utilizarían los algoritmos de reemplazo que se han visto en la sección precedente.

Con la política de asignación fija, es necesario decidir por adelantado la cantidad de espacio reservado para un proceso determinado. Esto se puede hacer en base al tipo de aplicación y al tamaño del programa.

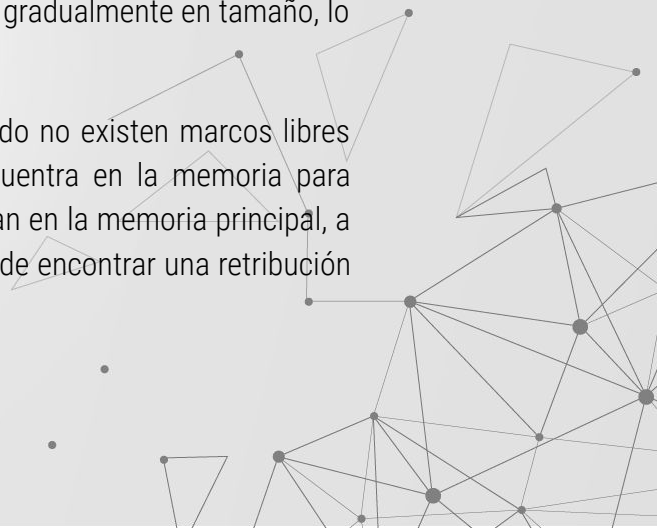


# Gestión del conjunto residente - Asignación variable, ámbito global

Esta combinación es, probablemente, la más sencilla de implementar y ha sido adoptada por un gran número de sistemas operativos.

En un momento determinado, existen un número de procesos en la memoria principal, donde cada uno tiene una serie de marcos asignados. Normalmente, el sistema operativo también mantiene una lista de marcos libres. Cuando sucede un fallo de página, se añade un marco libre al conjunto residente de un proceso y se trae la página a dicho marco. De esta forma, un proceso que sufra diversos fallos de página crecerá gradualmente en tamaño, lo cual debería reducir la tasa de fallos de página global del sistema.

La dificultad de esta estrategia se encuentra en la elección de los reemplazos cuando no existen marcos libres disponibles, el sistema operativo debe elegir una página que actualmente se encuentra en la memoria para reemplazarla. Esta selección se lleva a cabo entre todos los marcos que se encuentran en la memoria principal, a excepción de los marcos bloqueados como son los usados por el núcleo. Una forma de encontrar una retribución es el uso de buffering de páginas para aliviar el impacto en el rendimiento.



# Gestión del conjunto residente - Asignación variable, ámbito local

Intenta resolver los problemas de la estrategia de ámbito global.

1. Cuando se carga un nuevo proceso en la memoria principal, se le asignan un cierto número de marcos de página a su conjunto residente, basándose en el tipo de aplicación, solicitudes del programa, u otros criterios.
2. Cuando ocurra un fallo de página, la página que se seleccionará para reemplazar pertenecerá al conjunto residente del proceso que causó el fallo.
3. De vez en cuando, se reevaluará la asignación proporcionada a cada proceso, incrementándose o reduciéndose para mejorar el rendimiento.

Debido a la forma en la que se realiza esta valoración, esta estrategia es mucho más compleja que la política de reemplazo global simple. Sin embargo, puede llevar a un mejor rendimiento.



# Gestión del conjunto residente - Ámbito de reemplazo

Existe una correlación entre el ámbito de reemplazo y el tamaño del conjunto residente.

**Table 8.4.** Gestión del conjunto residente.

	Reemplazo Local	Reemplazo Global
<b>Asignación Fija</b>	<ul style="list-style-type: none"><li>• El número de marcos asignados a un proceso es fijo.</li><li>• Las páginas que se van a reemplazar se eligen entre los marcos asignados al proceso.</li></ul>	<ul style="list-style-type: none"><li>• No es posible</li></ul>
<b>Asignación Variable</b>	<ul style="list-style-type: none"><li>• El número de marcos asignados a un proceso pueden variarse de cuando en cuando.</li><li>• Las páginas que se van a reemplazar se eligen entre los marcos asignados al proceso.</li></ul>	<ul style="list-style-type: none"><li>• Las páginas que se van a reemplazar se eligen entre todos los marcos de la memoria principal esto hace que el tamaño del conjunto residente de los procesos varíe.</li></ul>

# Política de limpieza

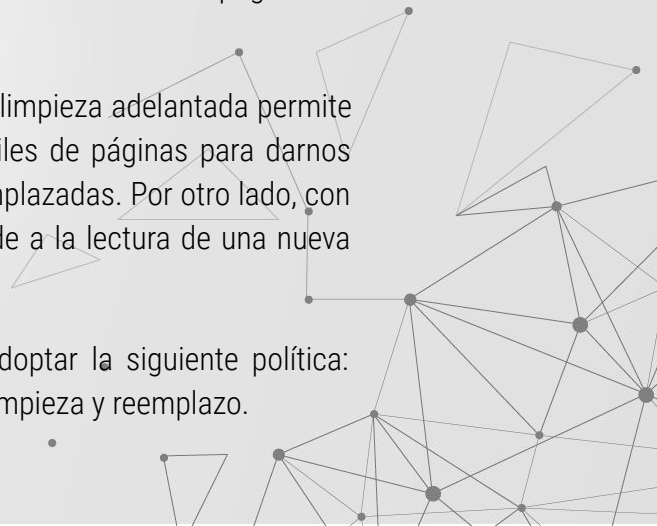
La política de limpieza es la opuesta a la política de recuperación; se encarga de determinar cuándo una página que está modificada se debe escribir en memoria secundaria. Las dos alternativas más comunes son la limpieza bajo demanda y la limpieza adelantada.

Con la *limpieza bajo demanda*, una página se escribe a memoria secundaria sólo cuando se ha seleccionado para un reemplazo.

En la política de la *limpieza adelantada* se escribe las páginas modificadas antes de que sus marcos de páginas se necesiten, de forma que las páginas se puedan escribir en lotes.

Existe un peligro en perseguir cualquiera de estas dos políticas hasta el extremo. La limpieza adelantada permite que las páginas se escriban en lotes, pero tiene poco sentido escribir cientos o miles de páginas para darnos cuenta de que la mayoría de ellas van a modificarse de nuevo antes de que sean reemplazadas. Por otro lado, con la limpieza bajo demanda, la escritura de una página modificada colisiona y procede a la lectura de una nueva página.

Una estrategia más apropiada es incorporar *buffering* de páginas. Esto permite adoptar la siguiente política: limpiar sólo las páginas que son reemplazables, pero desacoplar las operaciones de limpieza y reemplazo.

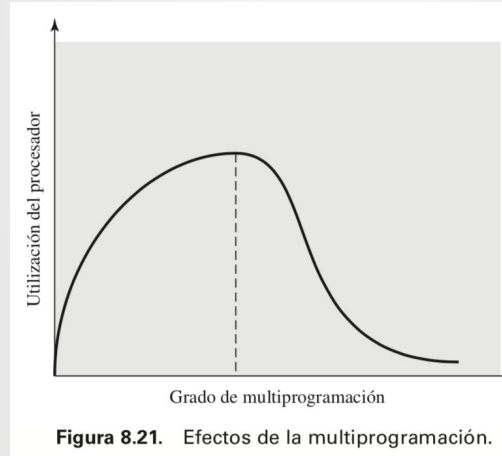


# Control de carga

El control de carga determina el número de procesos que residirán en la memoria principal, eso se denomina el grado de multiprogramación. La política de control de carga es crítica para una gestión de memoria efectiva.

Si hay muy pocos procesos residentes a la vez, habrá muchas ocasiones en las cuales todos los procesos se encuentren bloqueados, y gran parte del tiempo se gastará realizando *swapping*.

Por otro lado, si hay demasiados procesos residentes, entonces, de media, el tamaño de conjunto residente de cada proceso será poco adecuado y se producirán frecuentes fallos de página. El resultado es el trasiego (*thrashing*).



# Control de carga - suspensión de procesos

Si se va a reducir el grado de multiprogramación, uno o más de los procesos actualmente residentes deben suspenderse (enviarlos a swap). Se proporcionan seis posibilidades:

- Procesos con baja prioridad.
- Procesos que provocan muchos fallos.
- Proceso activado hace más tiempo.
- Proceso con el conjunto residente de menor tamaño.
- Proceso mayor.
- Proceso con la mayor ventana de ejecución restante.

