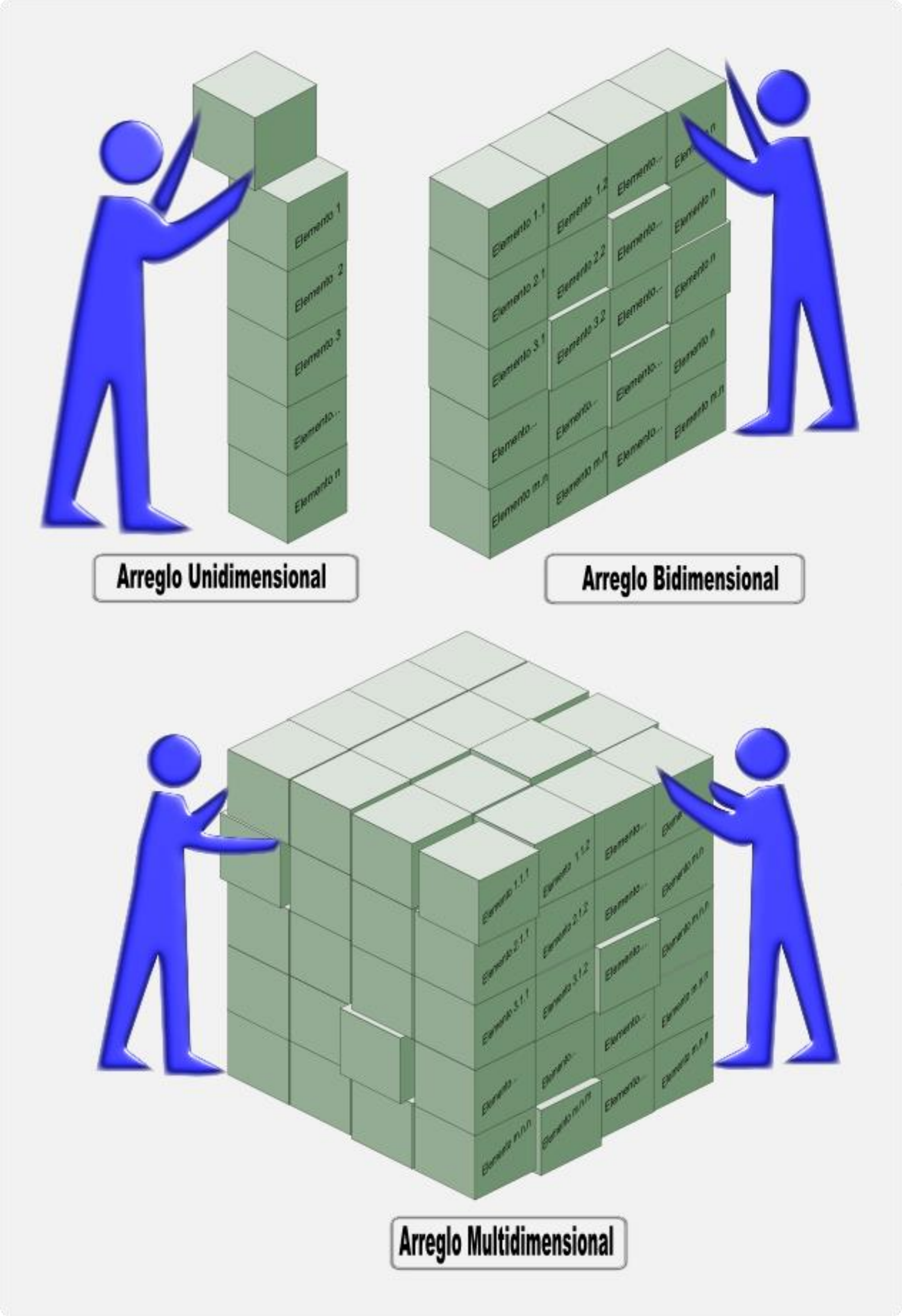


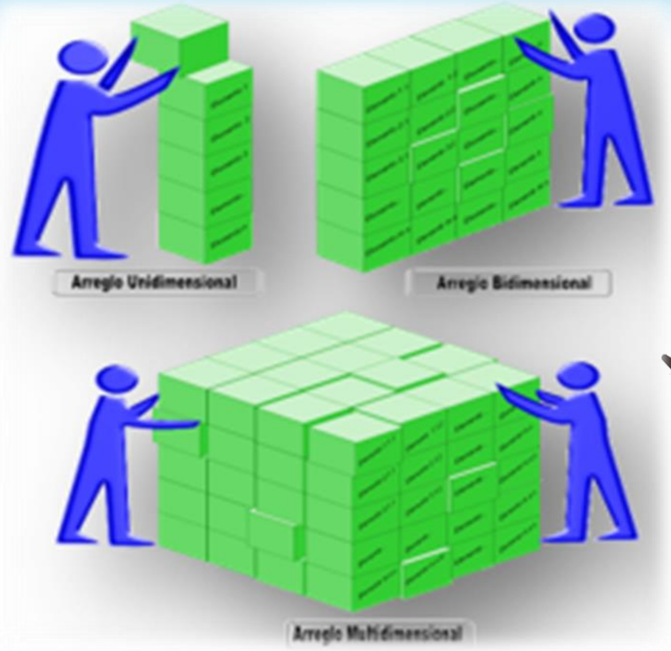
ESTRUCTURAS DE DATOS COMPUESTAS O ESTRUCTURADAS

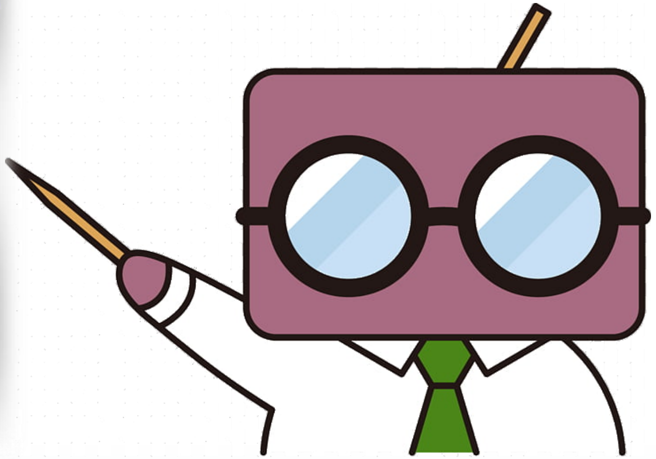
MÓDULO I. ESTRUCTURAS DE DATOS FUNDAMENTALES

INTRODUCCIÓN

La mayoría de los lenguajes de programación soporta diferentes estructuras de datos, es decir, posee una sintaxis propia para declararlas y funciones de biblioteca y operadores para manipularlas.

Algunos lenguajes, además, permiten a los programadores crear sus propias estructuras de datos con el objetivo fundamental de resolver de la forma más eficiente posible, una aplicación.





ESTRUCTURAS DE DATOS TIPO ARREGLOS

Arreglos

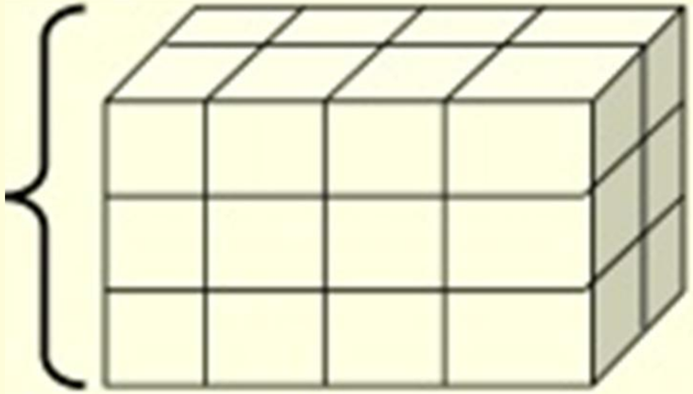
Las estructuras de datos pueden clasificarse en lineales y no lineales.

Se dice que una estructura es lineal si sus elementos forman una secuencia, que generalmente se almacenan en posiciones consecutivas de memoria.

Estas estructuras lineales reciben el nombre de arreglos (*arrays*) y pueden ser:

- Unidimensionales (también llamados vectores)
- Bidimensionales (matrices o tablas)
- Multidimensionales (tres o más dimensiones)

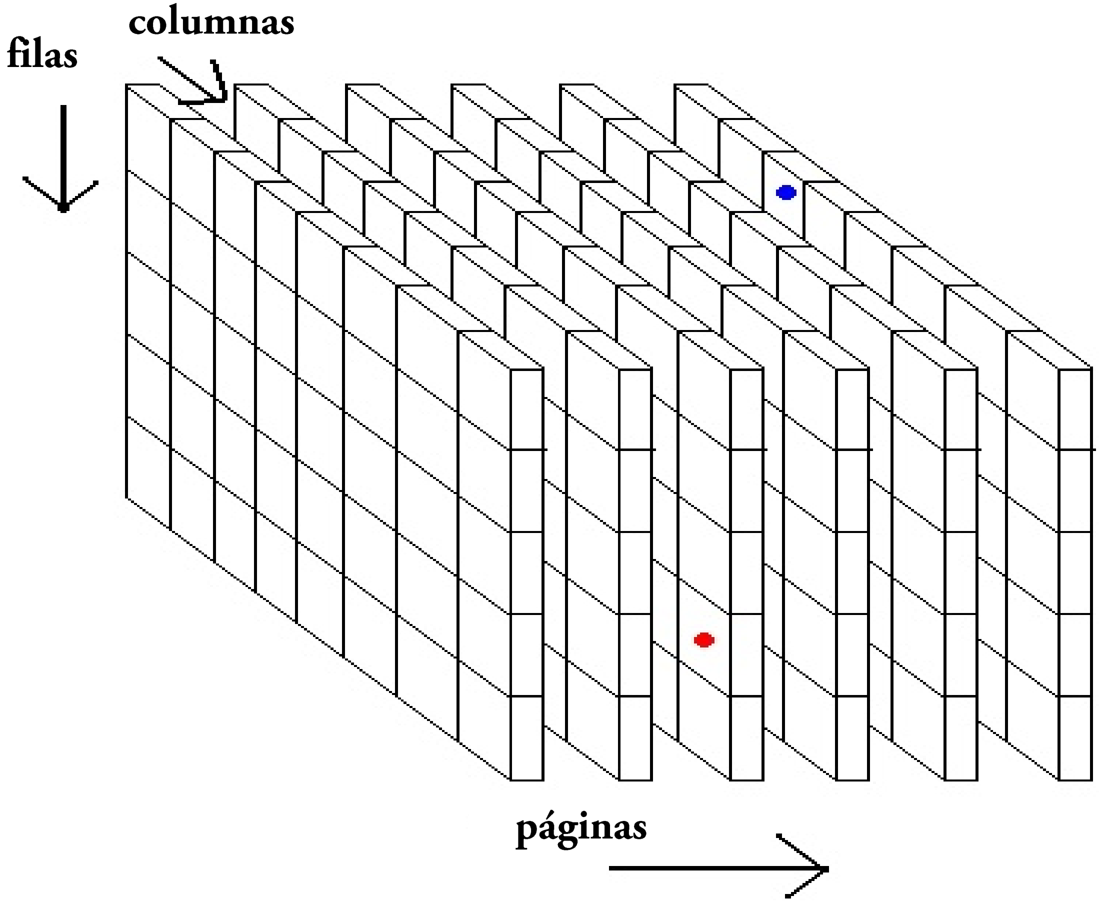
Arreglo de tres dimensiones o cubo. Los arreglos de dos o más dimensiones se les conoce como multidimensionales.



ARREGLOS MULTIDIMENSIONALES

Arreglos multidimensionales

Los arreglos multidimensionales se definen de forma análoga a los vistos hasta ahora, como un conjunto *finito* y *ordenado* de elementos *homogéneos*.



Arreglos multidimensionales

Concretamente, un arreglo *n-dimensional* **B** de $m_1 \; x \; m_2 \; x \cdots x \; m_n$ es una colección de $m_1 * m_2 * \cdots * m_n$ elementos en la que cada uno de ellos se especifica mediante una lista de n números enteros tales como K_1, K_2, ..., K_n, que reciben el nombre de ***índices.*** Estos n números tienen la propiedad de que

$$1 \leq K_1 \leq m_1 , \; 1 \leq K_2 \leq m_2 , \; ..., \; 1 \leq K_n \leq m_n$$

Al elemento de B con índices K_1, K_2, ..., K_n, lo simbolizaremos por

$$B[K_1, K_2, ..., K_n]$$

Arreglos Multidimensionales

Declaración de arreglos multidimensionales

Utilizaremos la siguiente sentencia para declarar estos arreglos

ident_arreglo: arreglo [LímInfF .. LímSupF, LímInfC .. LímSupC, LímInfP .. LímSupP, ..., LímInfN .. LímSupN] de tipo

Con **tipo** se declara el tipo de datos para todos los componentes del arreglo multidimensional.

Con los valores **LímInf y LímSup** de cada dimensión se declara el tipo de los índices, así como el número de elementos en cada dimensión.

Arreglos Multidimensionales

Los arreglos multidimensionales permiten también definir límite inferiores distintos de 1. La longitud L_i de la dimensión i de estos arreglos, puede calcularse mediante la fórmula

$$L_i = (LímSup - LímInf + 1)$$

Así, el ***NTC*** de estos arreglos será igual al producto de estas longitudes.

Para un índice K_i, determinado, el índice efectivo E_i de L_i, es el número de índices que pertenecen al conjunto de índices de la dimensión y que preceden a K_i. Este índice E_i lo podemos calcular

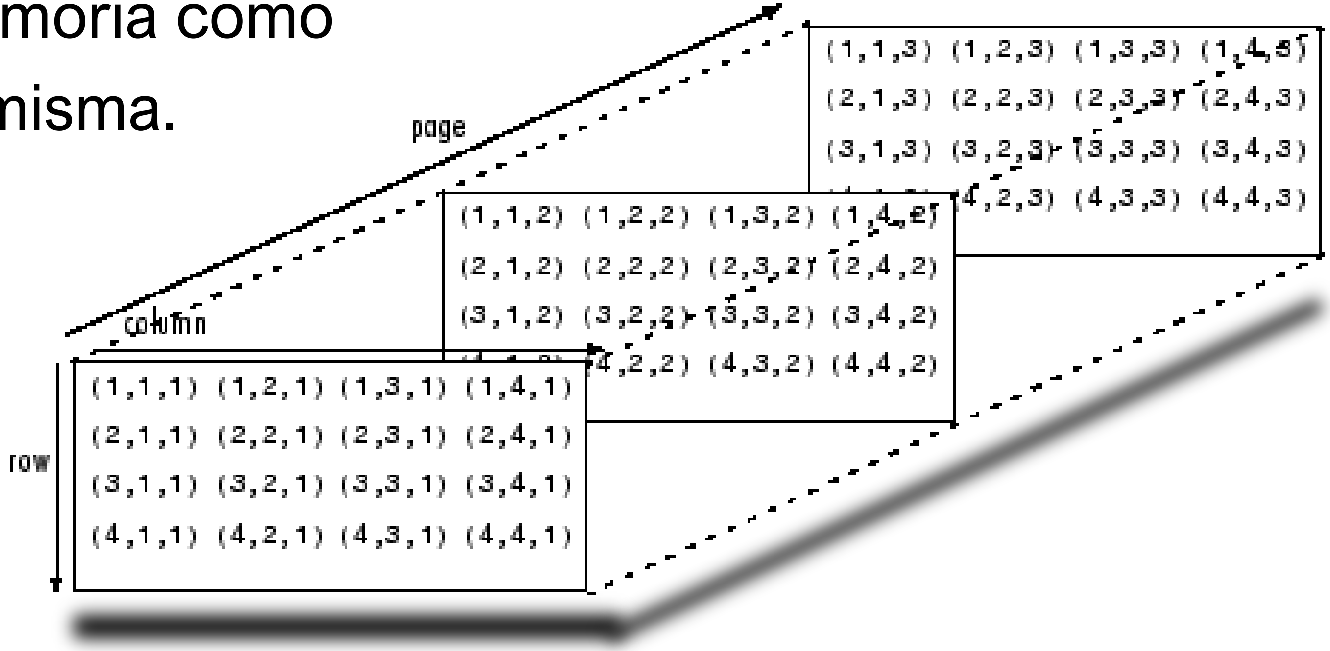
$$E_i = (K_i - LímInf)$$

Arreglos multidimensionales

Representación en memoria

El arreglo se almacenará en memoria como una secuencia de celdas de la misma.

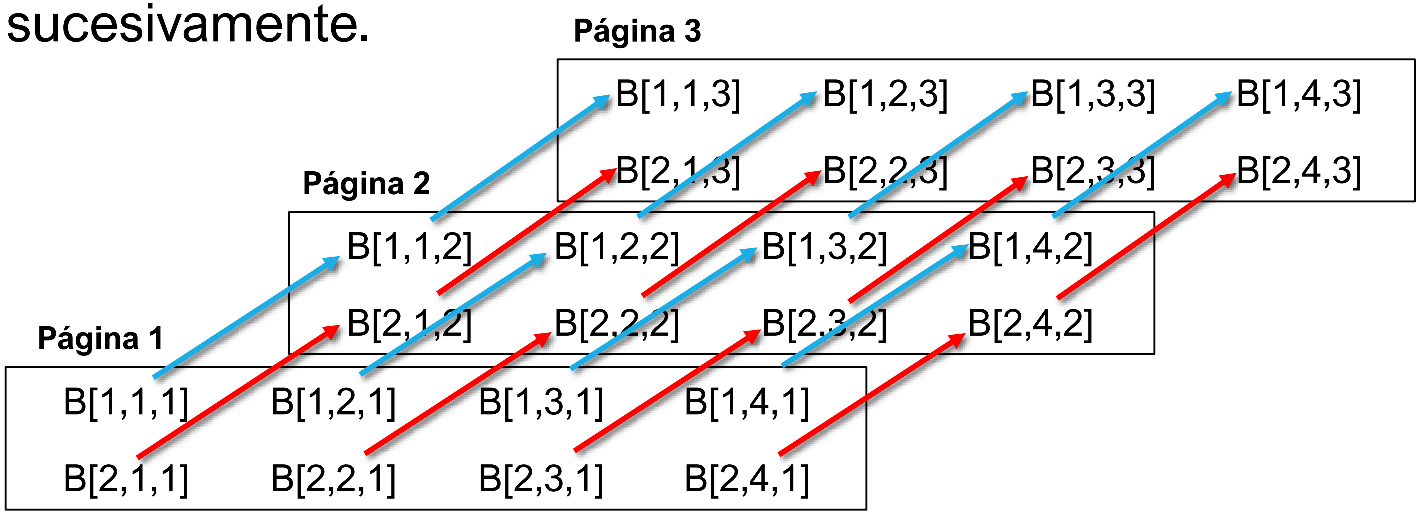
Concretamente, el lenguaje de programación puede colocarlos según la ordenación por columnas o la ordenación por filas.

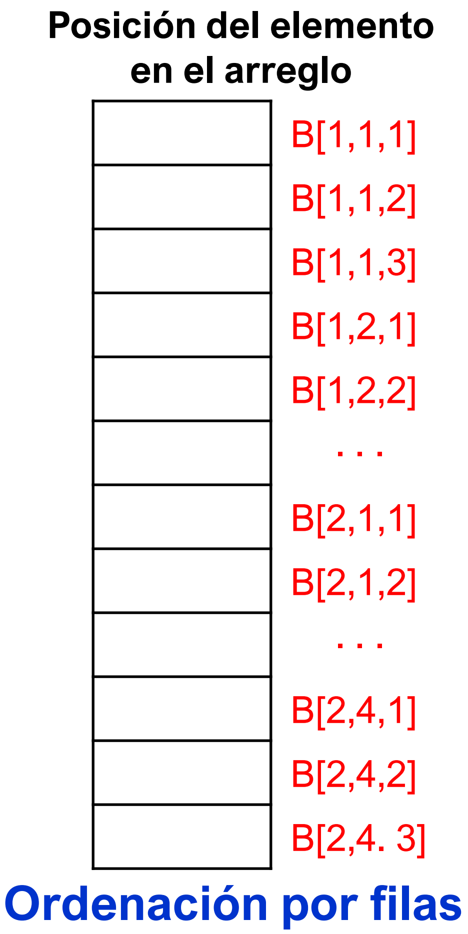


Arreglos multidimensionales

Representación en memoria

Para estos arreglos entendemos una ***ordenación por filas*** a aquella en que los elementos están ordenados de tal forma que los índices varían como el cuentakilómetros de un auto, es decir, el último índice varía primero (más rápidamente), el penúltimo índice varía segundo, (menos rápidamente), y así sucesivamente.

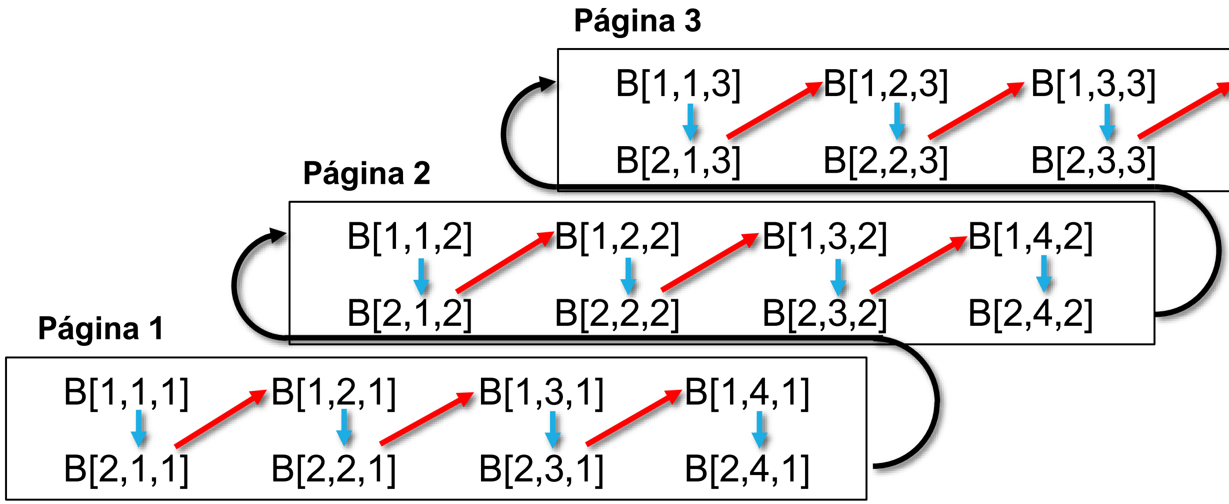


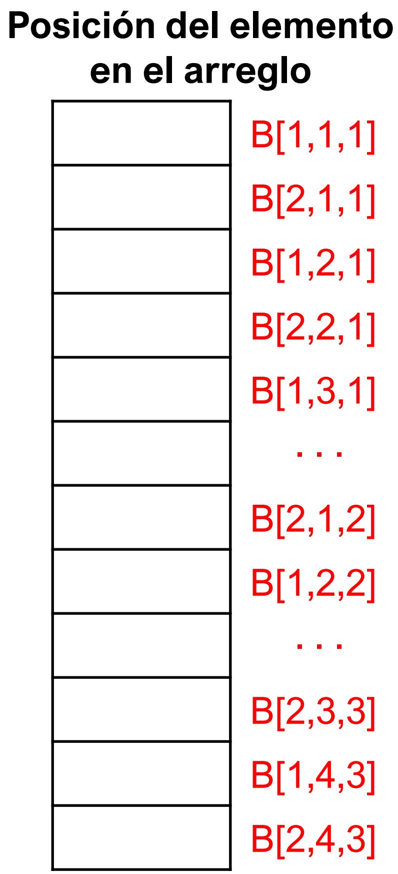


Arreglos multidimensionales

Representación en memoria

Por el contrario, en una ***ordenación por columnas*** los elementos se ordenan de tal forma que el primer índice varía primero (más rápidamente), el segundo índice varía después, (menos rápidamente), y así sucesivamente.





Ordenación por columnas

$L_1 = 7$ $E_1 = 3$
$L_2 = 6$ $E_2 = 3$
$L_3 = 5$ $E_3 = 2$

Arreglos multidimensionales

Representación en memoria

A: arreglo [2 .. 8, -4 .. 1, 6 .. 10] de enteros

Por tanto, la localización de un elemento arbitrario en uno de estos arreglos puede obtenerse de la fórmula

$$Loc(A[K_1, K_2, ..., K_n]) = dirBase(A) + w\,[(((\ldots(E_N L_{N-1} + E_{N-1})L_{N-2})+\ldots+\underline{E_3)L_2 + E_2)L_1 + E_1}]$$

O bien,

$$Loc(A[K_1, K_2, ..., K_n]) = dirBase(A) + w\,[(\ldots\underline{((E_1 L_2 + E_2)L_3 + E_3)}L_4+\ldots+ E_{N-1})L_N + E_N]$$

dependiendo de si almacenamos A ordenado por columnas o bien por filas.

Como siempre, ***dirBase***(**A**) indica la dirección del primer elemento y ***w*** el número de bytes que ocupa.

Arreglos multidimensionales

Representación en memoria

Ejemplo:

Suponga un arreglo tridimensional A, declarado como sigue

A: arreglo [2 .. 8, -4 .. 1, 6 .. 10] de enteros

Las longitudes de las tres dimensiones de A serán, respectivamente,

$L_1 = (8 - 2 + 1) = 7$, $L_2 = (1 - (-4) + 1) = 6$, $L_3 = (10 - 6 + 1) = 5$

Por tanto,

$$\text{NTC(A)} = L_1 * L_2 * L_3 = 7 * 6 * 5 = 210 \text{ elementos}$$

Arreglos multidimensionales

Representación en memoria

EJEMPLO (CONT.):

A: arreglo [2 .. 8, -4 .. 1, 6 .. 10] de enteros

Supongamos ahora que el arreglo tridimensional A, se almacena ordenado por filas y que, ***dirBase***(A) = 1000 y ***w*** = 4 bytes por celda de memoria.

La dirección del elemento A[5, -1, 8] la podemos obtener de la forma siguiente:

a) *Obtenemos los índice efectivos de acuerdo a la fórmula* $E_i = (K_i - LímInf)$

$E_1 = (5 - 2) = 3$, $E_2 = (-1) - (-4) = 3$, $E_3 = (8 - 6) = 2$

b) *Se aplica la fórmula* $Loc(A[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1 L_2 + E_2) L_3 + E_3)]$

Arreglos multidimensionales

Representación en memoria

Ejemplo (cont.):

Así, reemplazamos los valores obtenidos previamente en la fórmula dada,

$L_1 = 7$	$E_1 = 3$
$L_2 = 6$	$E_2 = 3$
$L_3 = 5$	$E_3 = 2$

$$Loc(A[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1L_2 + E_2)L_3 + E_3)]$$

Arreglos multidimensionales

Representación en memoria

EJEMPLO (CONT.):

$L_1 = 7$ $E_1 = 3$
$L_2 = 6$ $E_2 = 3$
$L_3 = 5$ $E_3 = 2$

Así, reemplazamos los valores obtenidos previamente en la fórmula dada,

$$Loc(A[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1L_2 + E_2)L_3 + E_3)]$$

$$Loc(A[5, -1, 8]) = 1000 + 4\,[\,(3 * 6 + 3) * 5 + 2)]$$

Arreglos multidimensionales

Representación en memoria

EJEMPLO (CONT.):

Así, reemplazamos los valores obtenidos previamente en la fórmula dada,

$L_1 = 7 \quad E_1 = 3$
$L_2 = 6 \quad E_2 = 3$
$L_3 = 5 \quad E_3 = 2$

$$Loc(\mathrm{A}[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1L_2 + E_2)L_3 + E_3)]$$

$$Loc(\mathrm{A}[5, -1, 8]) = 1000 + 4\,[\,(3 * 6 + 3) * 5 + 2)]$$

$$= 1000 + 4\,[\,(18 + 3) * 5 + 2)]$$

Arreglos multidimensionales

Representación en memoria

EJEMPLO (CONT.):

Así, reemplazamos los valores obtenidos previamente en la fórmula dada,

$L_1 = 7$ $E_1 = 3$
$L_2 = 6$ $E_2 = 3$
$L_3 = 5$ $E_3 = 2$

$$Loc(\mathrm{A}[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1L_2 + E_2)L_3 + E_3)]$$

$$\begin{aligned} Loc(\mathrm{A}[5, -1, 8]) &= 1000 + 4\,[\,(3 * 6 + 3) * 5 + 2)] \\ &= 1000 + 4\,[\,(18 + 3) * 5 + 2)] \\ &= 1000 + 4\,[\,(21) * 5 + 2)] \end{aligned}$$

Arreglos multidimensionales

Representación en memoria

EJEMPLO (CONT.):

$L_1 = 7$	$E_1 = 3$
$L_2 = 6$	$E_2 = 3$
$L_3 = 5$	$E_3 = 2$

Así, reemplazamos los valores obtenidos previamente en la fórmula dada,

$$Loc(\mathrm{A}[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1L_2 + E_2)L_3 + E_3)]$$

$$Loc(\mathrm{A}[5, -1, 8]) = 1000 + 4\,[\,(3 * 6 + 3) * 5 + 2)]$$

$$= 1000 + 4\,[\,(18 + 3) * 5 + 2)]$$

$$= 1000 + 4\,[\,(21) * 5 + 2)]$$

$$= 1000 + 4\,[\,105 + 2)]$$

Arreglos multidimensionales

Representación en memoria

EJEMPLO (CONT.):

$L_1 = 7$ $E_1 = 3$
$L_2 = 6$ $E_2 = 3$
$L_3 = 5$ $E_3 = 2$

Así, reemplazamos los valores obtenidos previamente en la fórmula dada,

$$Loc(\mathrm{A}[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1L_2 + E_2)L_3 + E_3)]$$

$$\begin{aligned} Loc(\mathrm{A}[5, -1, 8]) &= 1000 + 4\,[\,(3 * 6 + 3) * 5 + 2)] \\ &= 1000 + 4\,[\,(18 + 3) * 5 + 2)] \\ &= 1000 + 4\,[\,(21) * 5 + 2)] \\ &= 1000 + 4\,[\,105 + 2)] \\ &= 1000 + 4\,[107] \end{aligned}$$

Arreglos multidimensionales

Representación en memoria

EJEMPLO (CONT.):

$L_1 = 7$ $E_1 = 3$
$L_2 = 6$ $E_2 = 3$
$L_3 = 5$ $E_3 = 2$

Así, reemplazamos los valores obtenidos previamente en la fórmula dada,

$$Loc(\mathrm{A}[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1L_2 + E_2)L_3 + E_3)]$$

$$\begin{aligned}
Loc(\mathrm{A}[5, -1, 8]) &= 1000 + 4\,[\,(3 * 6 + 3)* 5 + 2)] \\
&= 1000 + 4\,[\,(18 + 3)* 5 + 2)] \\
&= 1000 + 4\,[\,(21)* 5 + 2)] \\
&= 1000 + 4\,[\,105 + 2)] \\
&= 1000 + 4\,[107] \\
&= 1000 + 428
\end{aligned}$$

Arreglos multidimensionales

Representación en memoria

EJEMPLO (CONT.):

$L_1 = 7$ $E_1 = 3$
$L_2 = 6$ $E_2 = 3$
$L_3 = 5$ $E_3 = 2$

Así, reemplazamos los valores obtenidos previamente en la fórmula dada,

$$Loc(A[K_1, K_2, K_3]) = dirBase(A) + w\,[\,(E_1L_2 + E_2)L_3 + E_3)]$$

$$\begin{aligned}
Loc(A[5, -1, 8]) &= 1000 + 4\,[\,(3 * 6 + 3)* 5 + 2)] \\
&= 1000 + 4\,[\,(18 + 3)* 5 + 2)] \\
&= 1000 + 4\,[\,(21)* 5 + 2)] \\
&= 1000 + 4\,[\,105 + 2)] \\
&= 1000 + 4\,[107] \\
&= 1000 + 428
\end{aligned}$$

$$Loc(A[5, -1, 8]) = 1428$$

Arreglos multidimensionales

Representación en memoria

PRÁCTICA:

Asuma el arreglo A del ejemplo, con ***dirBase*(A)** = 1000 y ***w*** = **4** (**enteros**), obtenga:

(a) **LOC(A[5, -1, 8])**, asumiendo que el arreglo fue almacenado columna a columna.

(b) **LOC(A[2, 0, 9])**, asumiendo que el arreglo fue almacenado fila a fila.

(c) **LOC(A[2, -4, 6])**, asumiendo que el arreglo fue almacenado fila a fila.

(d) **LOC(A[2, 0, 9])**, asumiendo que el arreglo fue almacenado columna a columna.

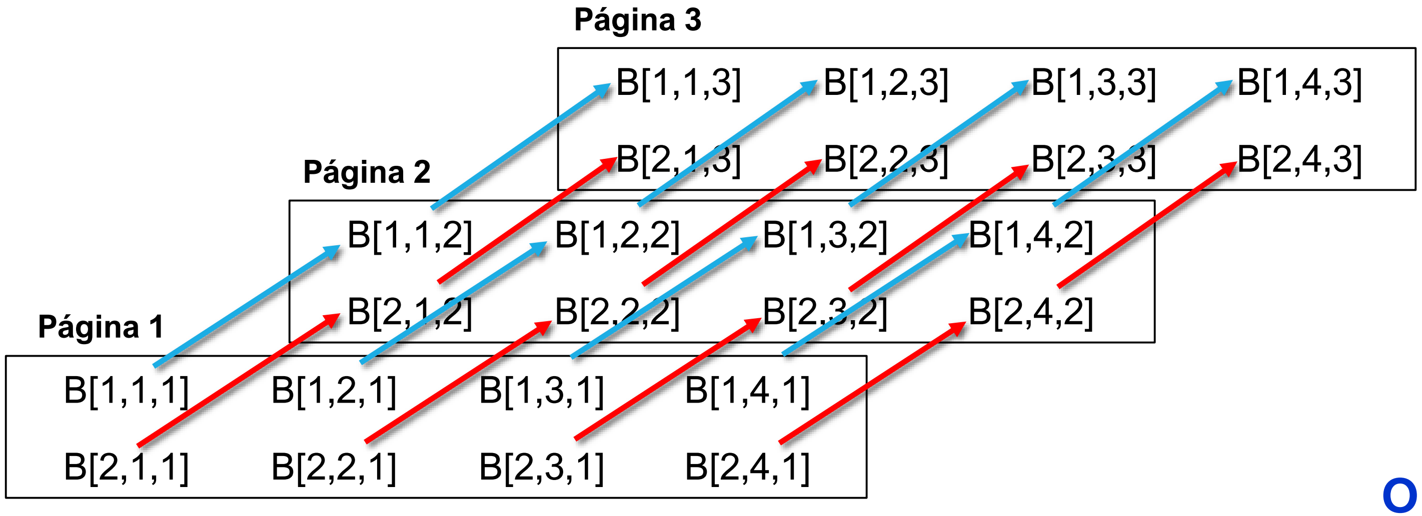
(e) **LOC(A[8, 1, 10)**, asumiendo que el arreglo fue almacenado columna a columna.

Arreglos multidimensionales

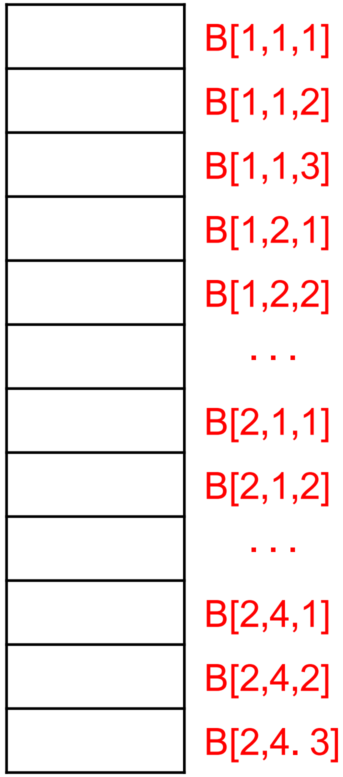
Operaciones sobre estos arreglos

Lectura: fila a fila

Mencionamos que en estos arreglos entendemos una ***ordenación por filas*** a aquella en que los elementos están ordenados de tal forma que los índices varían como el cuentakilómetros de un auto.



Posición del elemento en el arreglo



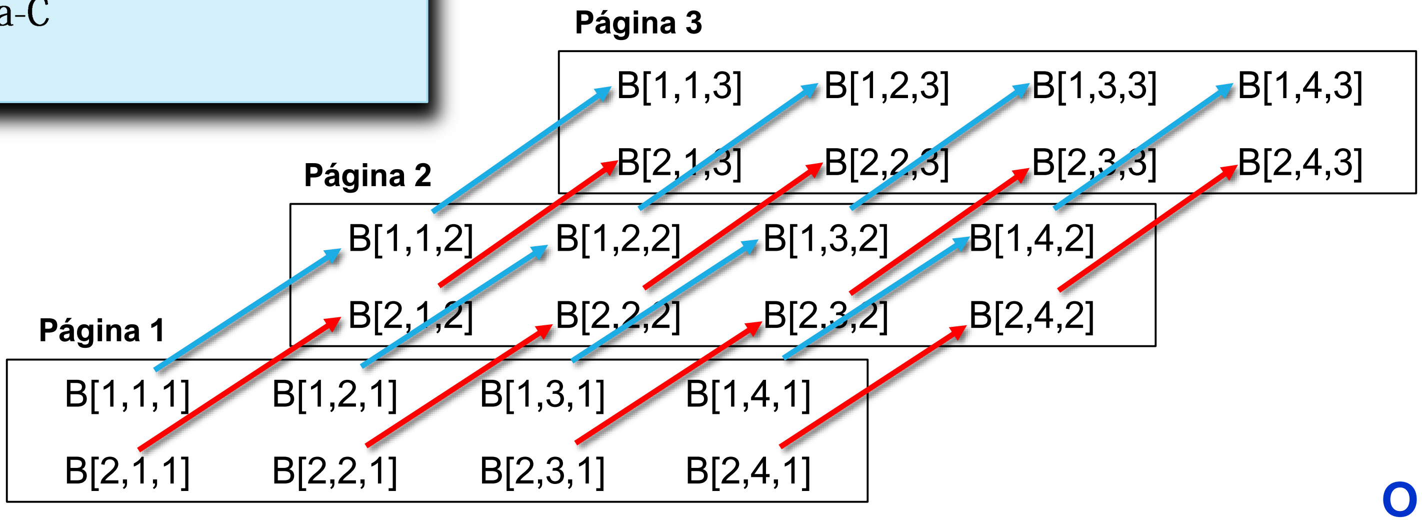
Ordenación por filas

Arreglos multidimensionales

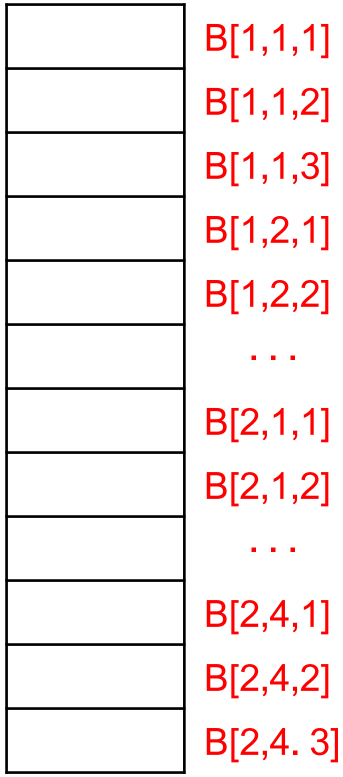
Operaciones sobre estos arreglos

Lectura: fila a fila

```
Para F =1, M, 1 repetir
    Para C = 1, N, 1 repetir
        Para P= 1, K, 1 repetir
            LEER (A[F,C,P])
        Fin-para-P
    Fin-para-C
Fin-para-F
```



Posición del elemento en el arreglo



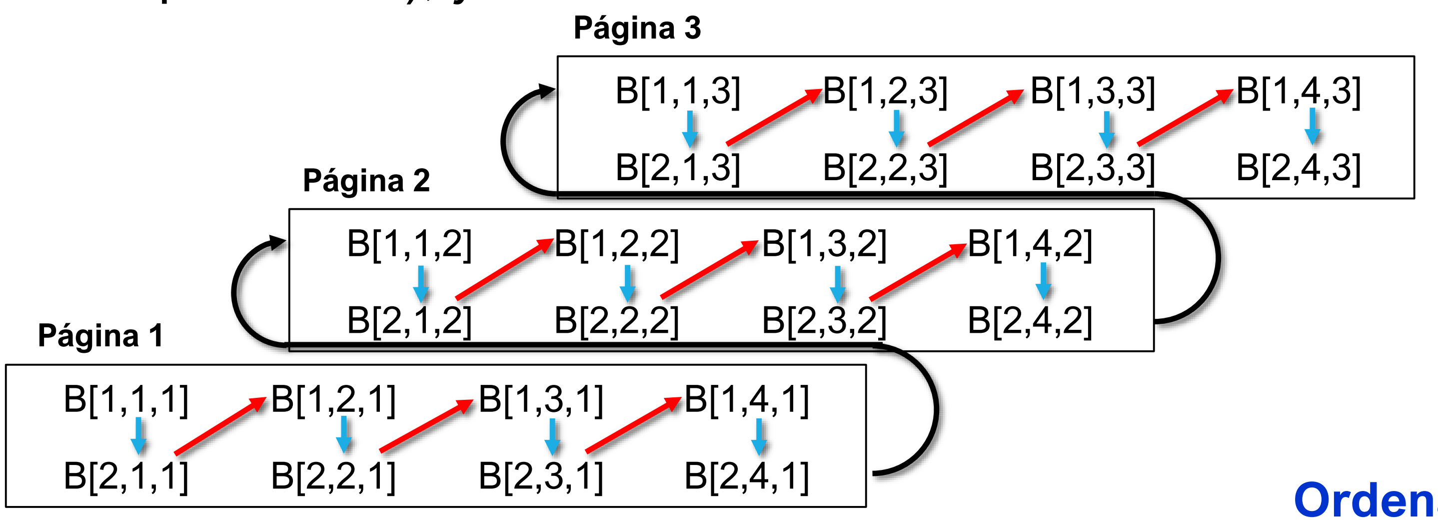
Ordenación por filas

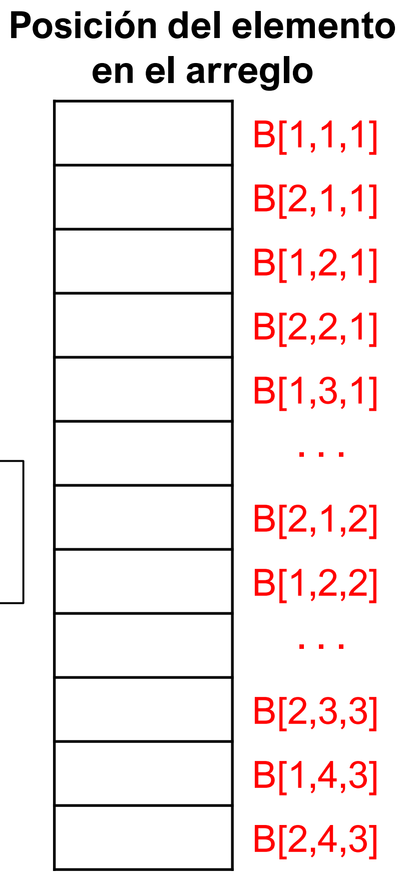
Arreglos multidimensionales

Operaciones sobre estos arreglos

Lectura: columna a columna

En la lectura con ***ordenación por columnas*** los elementos se ordenan de tal forma que el primer índice varía primero (más rápidamente), el segundo índice varía después, (menos rápidamente), y así sucesivamente.





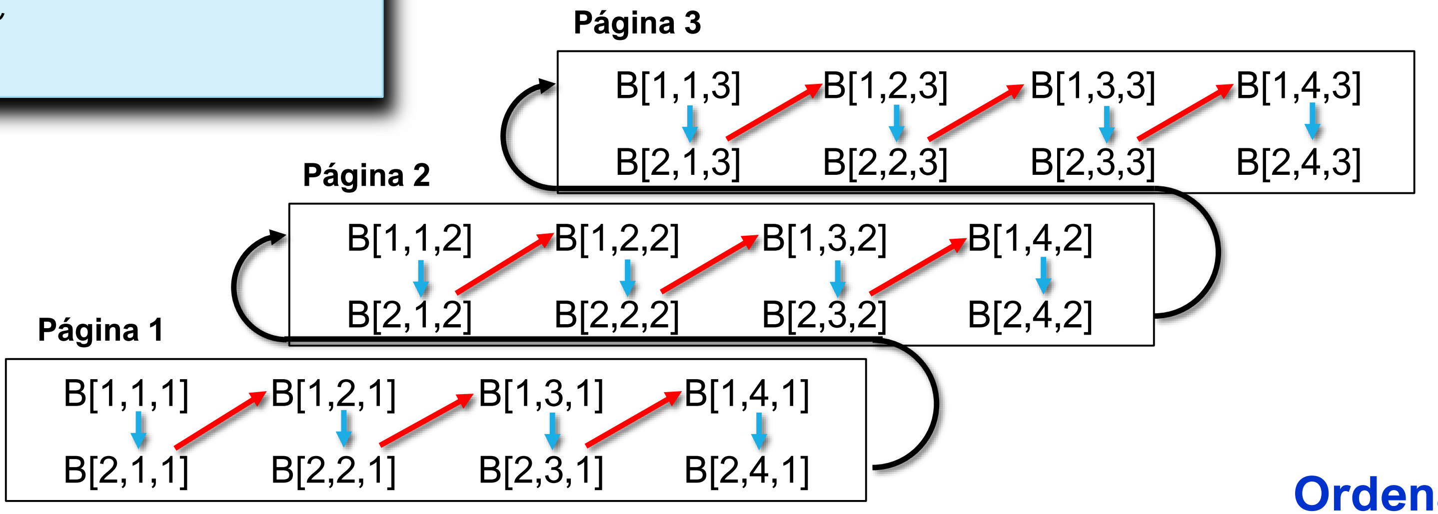
Ordenación por columnas

Arreglos multidimensionales

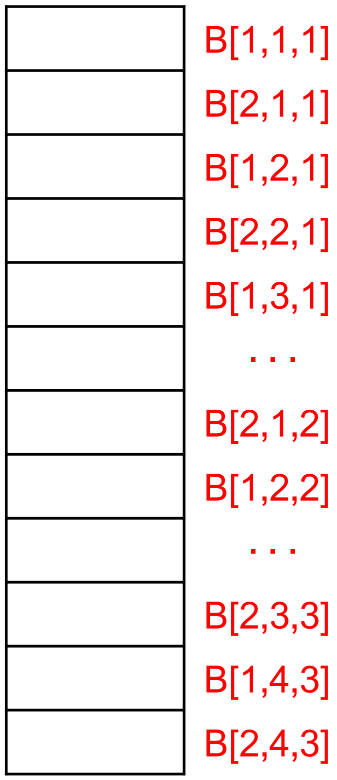
Operaciones sobre estos arreglos

Lectura: columna a columna

```
Para P =1, K, 1 repetir
    Para C = 1, N, 1 repetir
        Para F= 1, M, 1 repetir
            LEER (A[F,C,P])
        Fin-para-F
    Fin-para-C
Fin-para-P
```



Posición del elemento en el arreglo



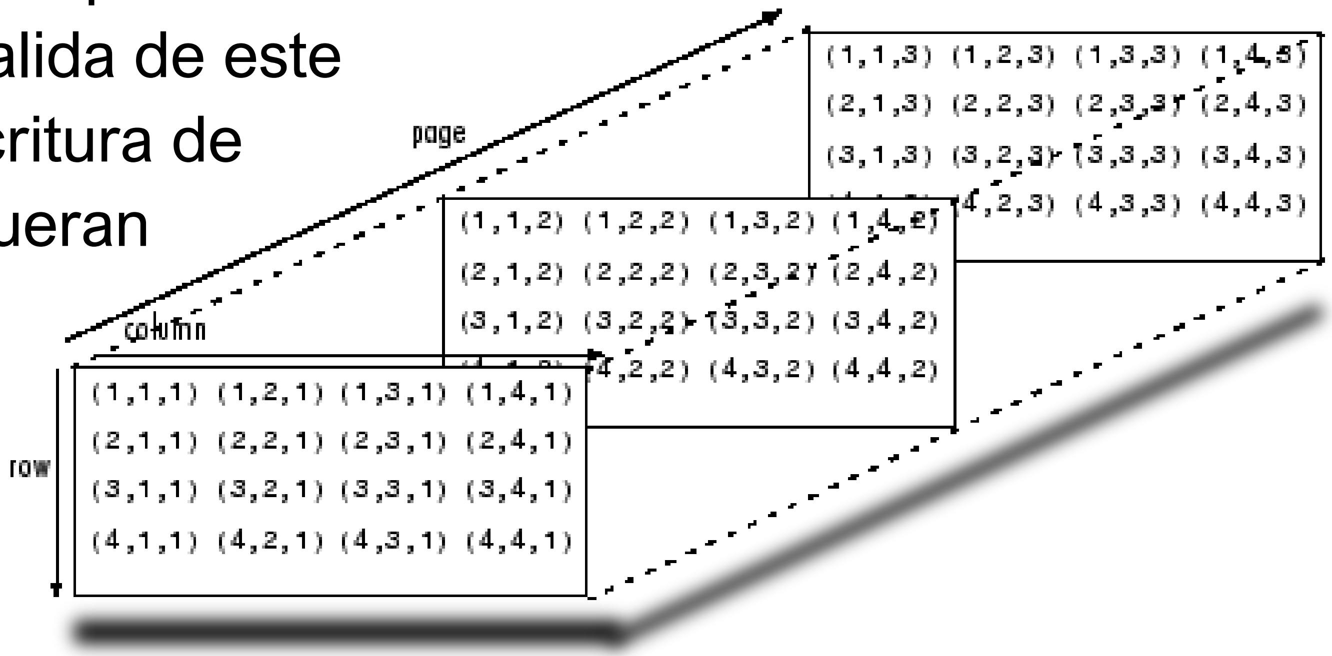
Ordenación por columnas

Arreglos multidimensionales

Operaciones sobre estos arreglos

Impresión del arreglo

Obviamente, el despliegue o la impresión son mecanismos 2D, por tanto la salida de este tipo de arreglos requiere la escritura de páginas individuales, como si fueran matrices o arreglos de dos dimensiones. Esto implica que el índice de página varíe más lentamente, luego el de fila y finalmente el de columnas.

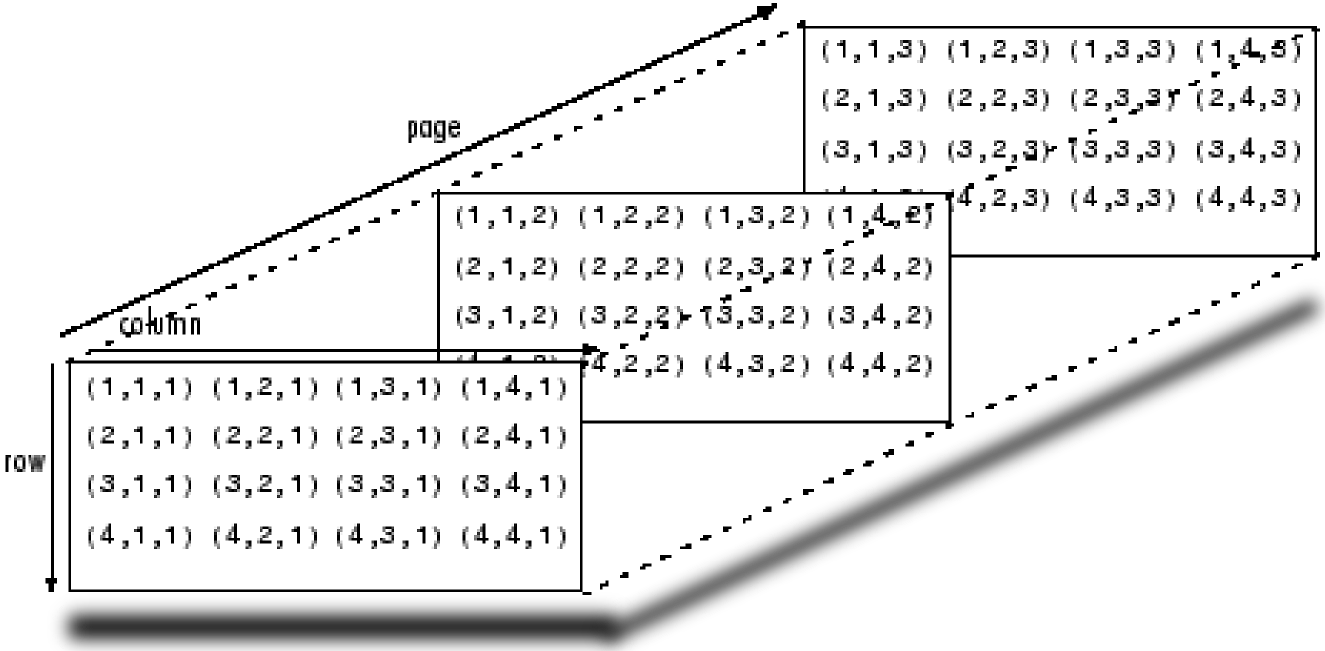


Arreglos multidimensionales

Operaciones sobre estos arreglos

Impresión del arreglo

```
Para P =1, K, 1 repetir
    Para F = 1, M, 1 repetir
        Para C= 1, N, 1 repetir
            ESCRIBIR (A[F,C,P])
        Fin-para-C
    Fin-para-F
Fin-para-P
```



Continuaremos con ... Registros...