Unidad 4 – Matriz Dispersa

Iintroducción a la Programación y Computación 2

Matriz

Una matriz o **arreglo bidimensional** ,es una zona de almacenamiento continuo, que contiene una serie de elementos del mismo tipo, desde el punto de vista lógico una matriz puede verse como un conjunto de elementos en fila, o filas y columnas si tuviesen dos dimensiones.

Cada elemento de la estructura puede ser accedida por medio del índice o posición del elemento en la matriz

Conceptos

- Es un arreglo con dos dimensiones de datos.
- La cantidad de columnas y filas puede o no ser la misma.
- En computación, es generalmente representada por arreglos.

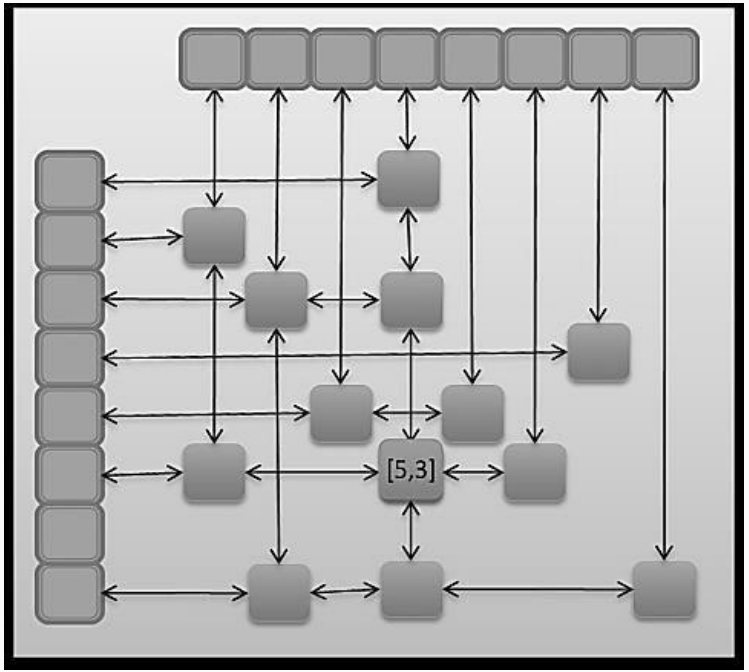
Matriz ortogonal

- Matriz MxN elementos.
- Es considerado como un caso especial de una matriz dispersa.
- Generalmente utilizada cuando todos los elementos tienen un valor.
- La forma más eficiente de implementarlo, en cuestión de acceso, es mediante un arreglo.

Matriz dispersa

Matriz en la cual la mayoría de sus elementos son 0.

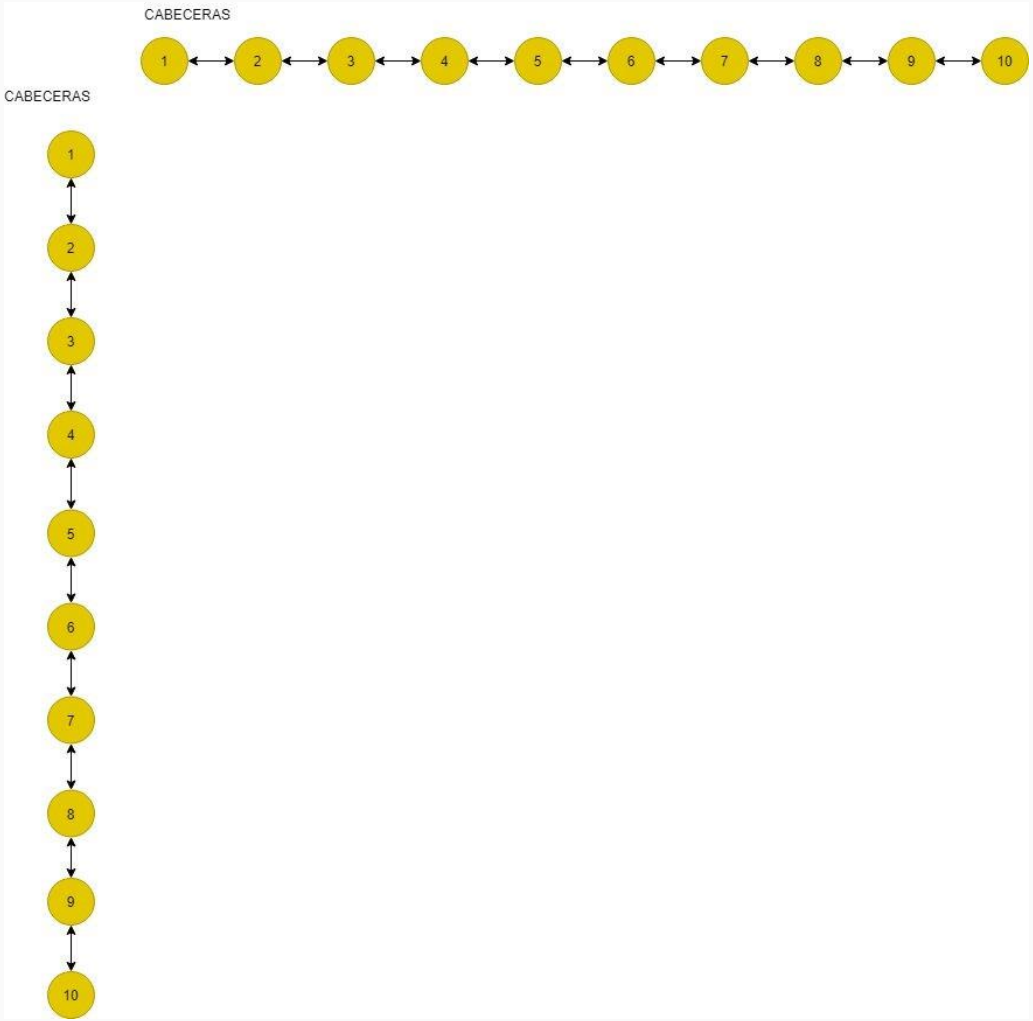
Existe un factor de dispersión que indica el % de valores 0 de la matriz.



Cabeceras

Cabeceras en X

Cabeceras en Y

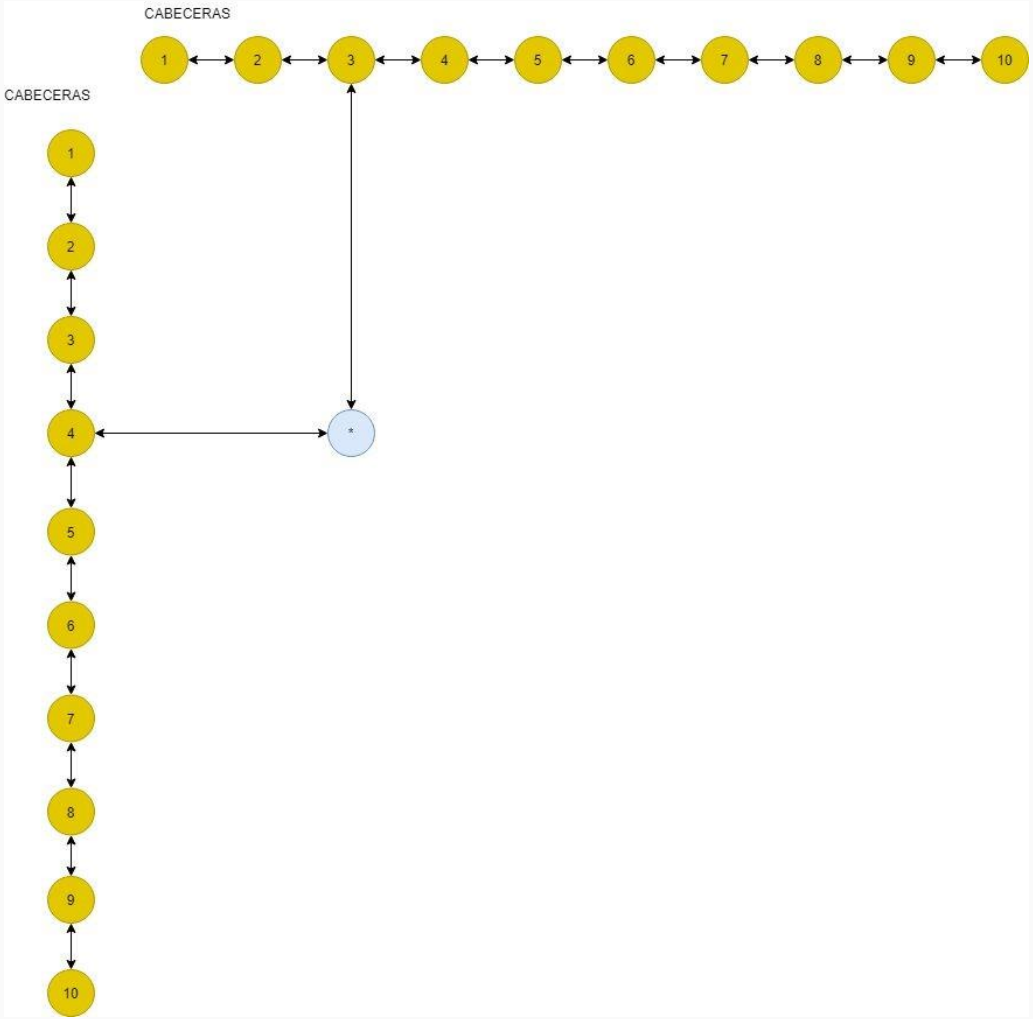


Insertar:

X: 3

Y: 4

Valor: *

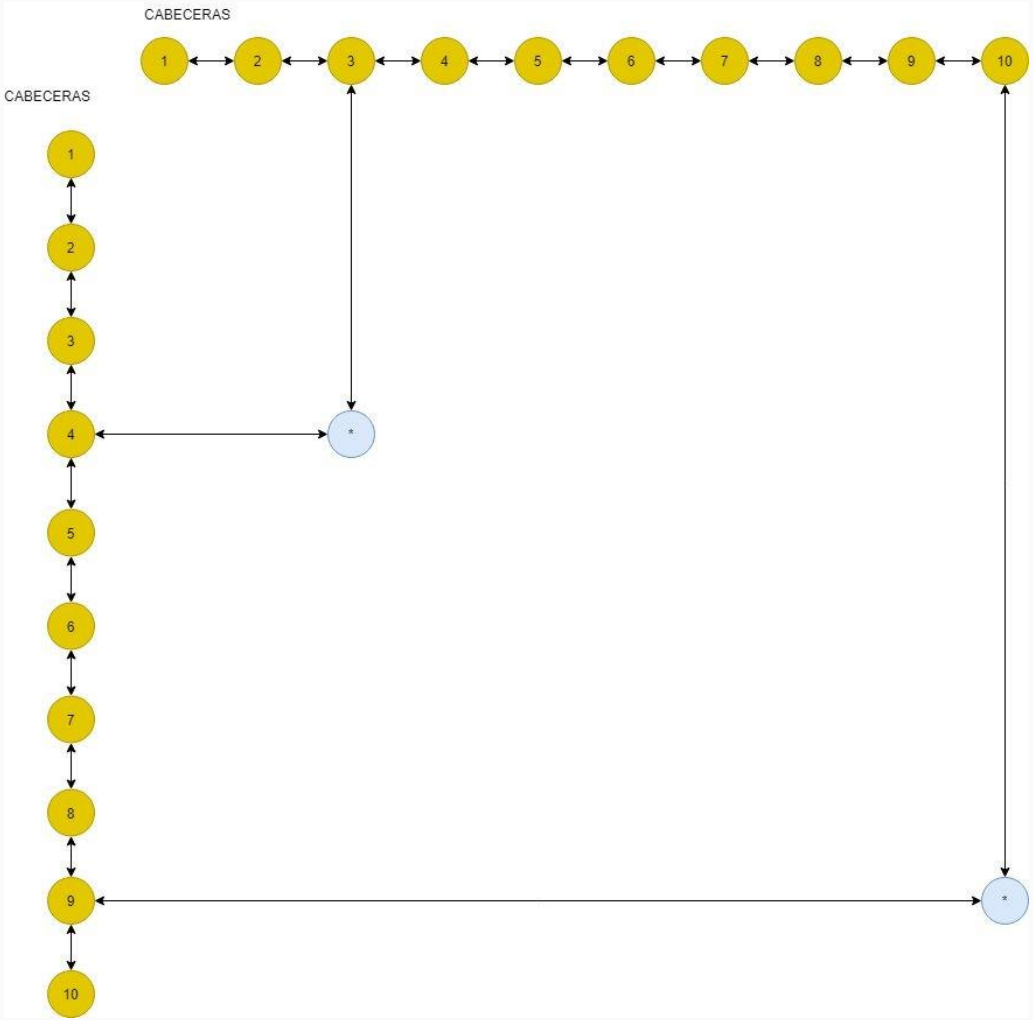


Insertar:

X: 10

Y: 9

Valor: *

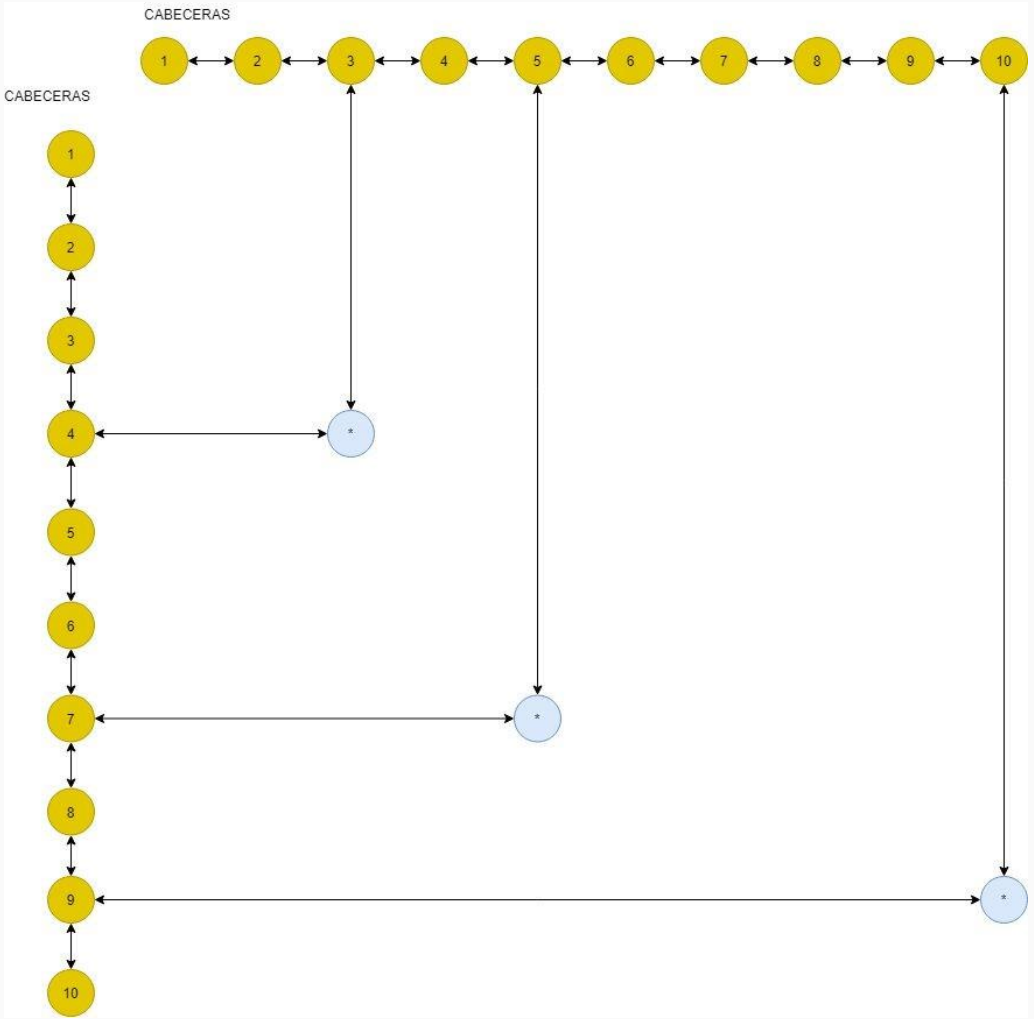


Insertar:

X: 5

Y: 7

Valor: *

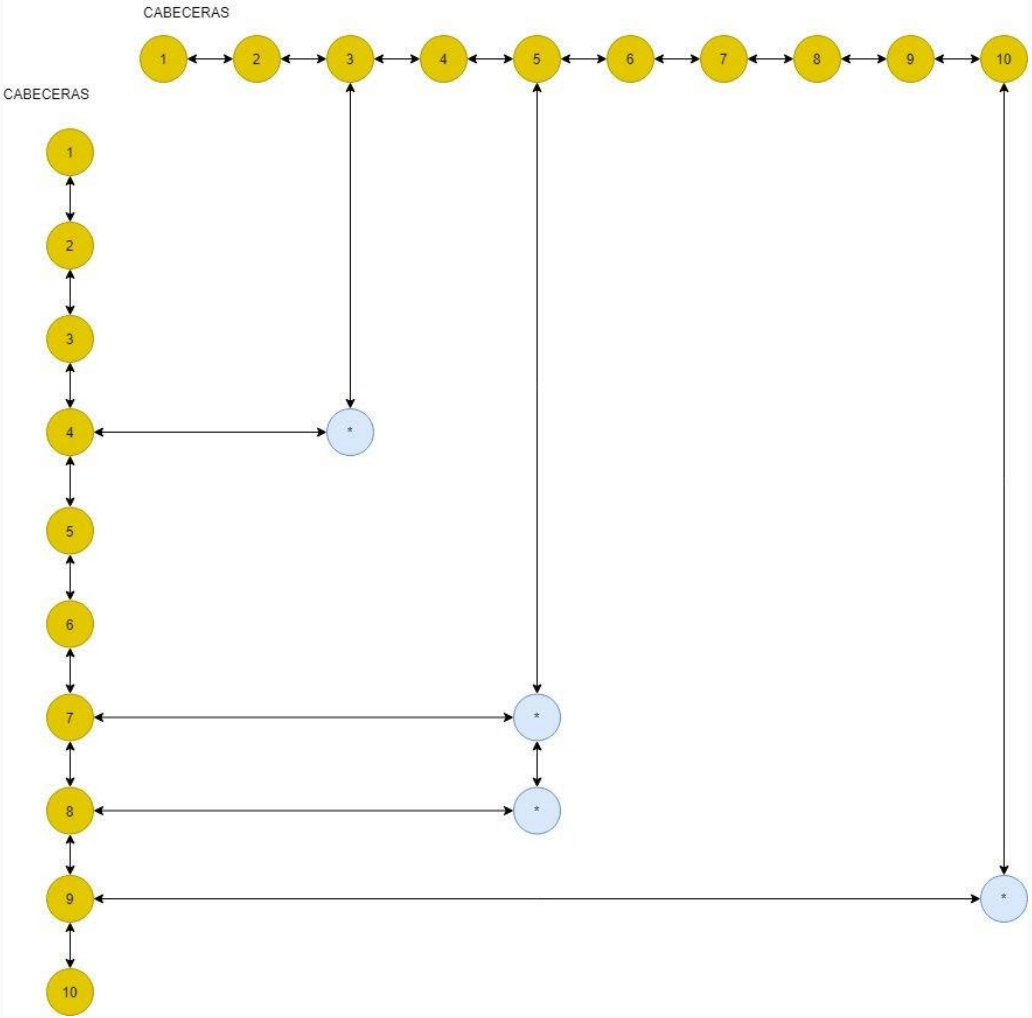


Insertar:

X: 5

Y: 8

Valor: *

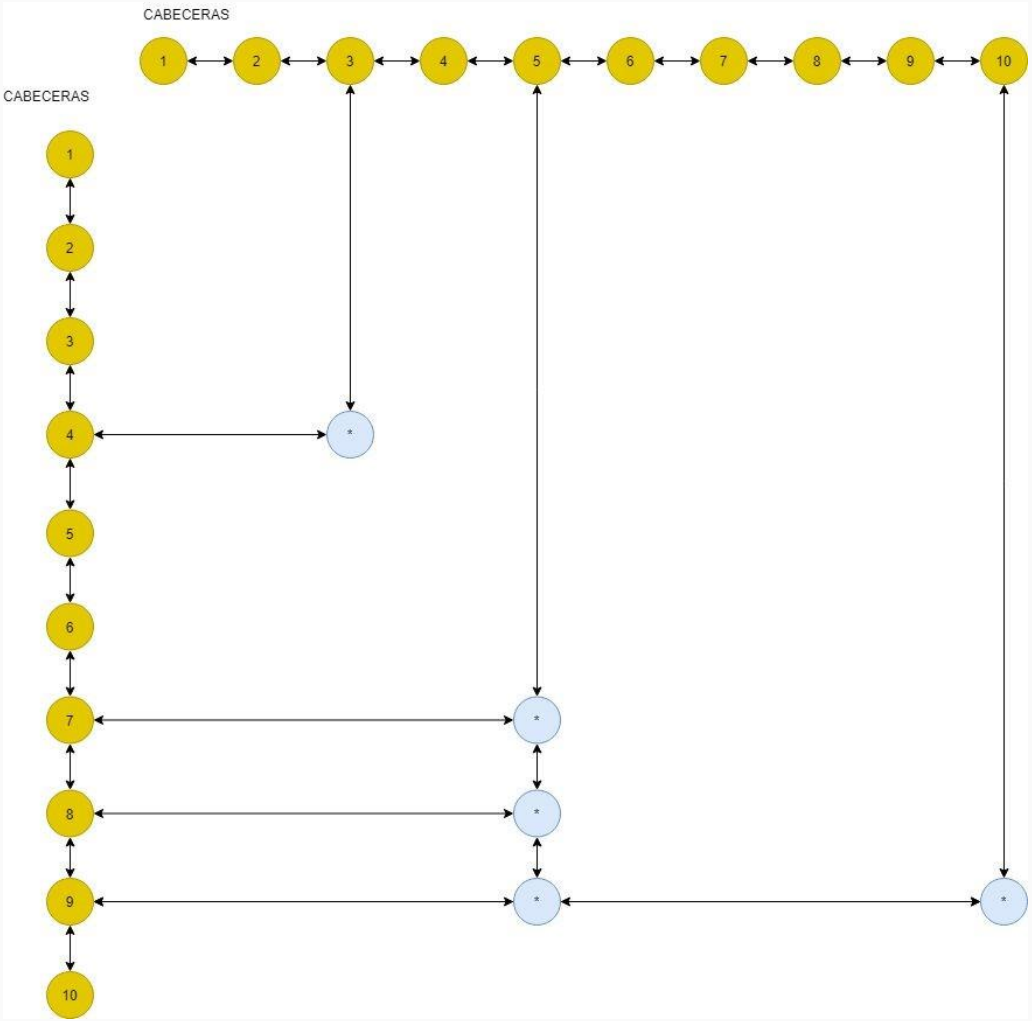


Insertar:

X: 5

Y: 9

Valor: *

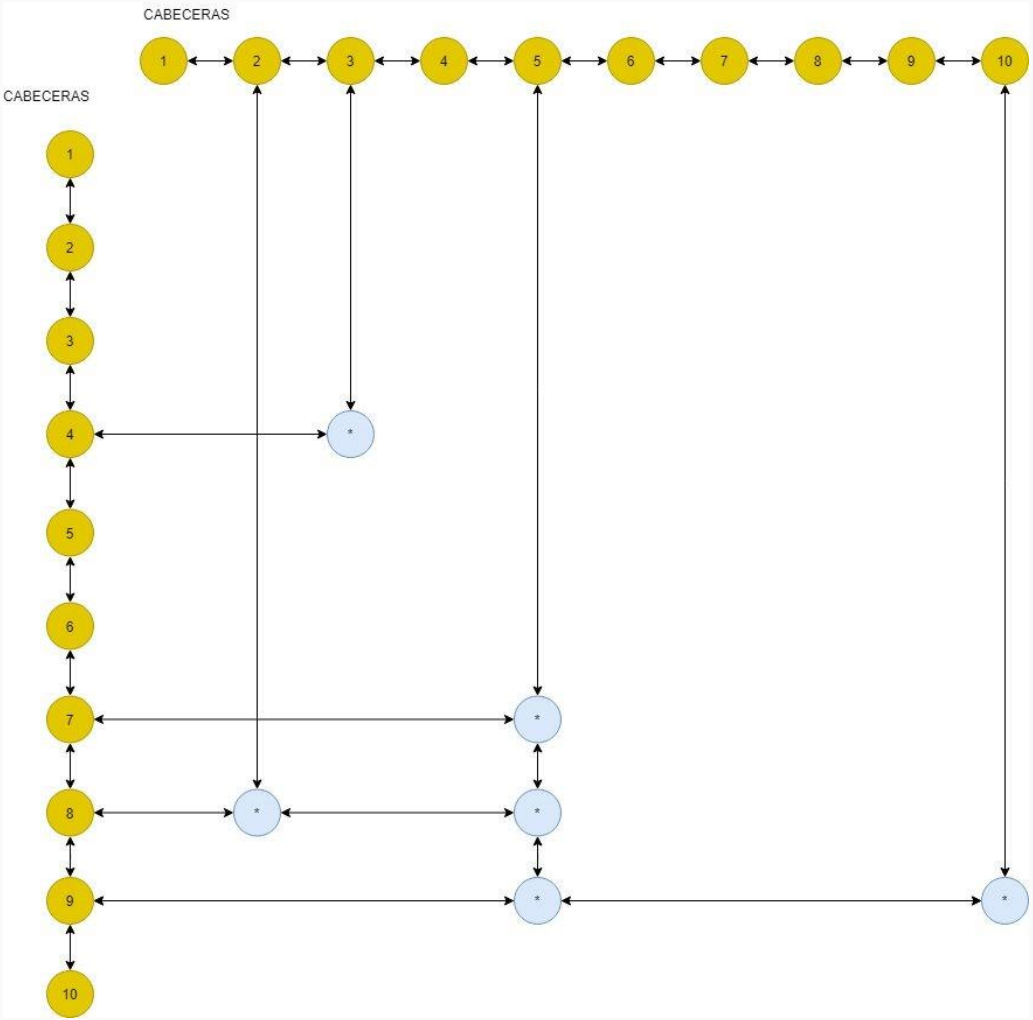


Insertar:

X: 2

Y: 8

Valor: *

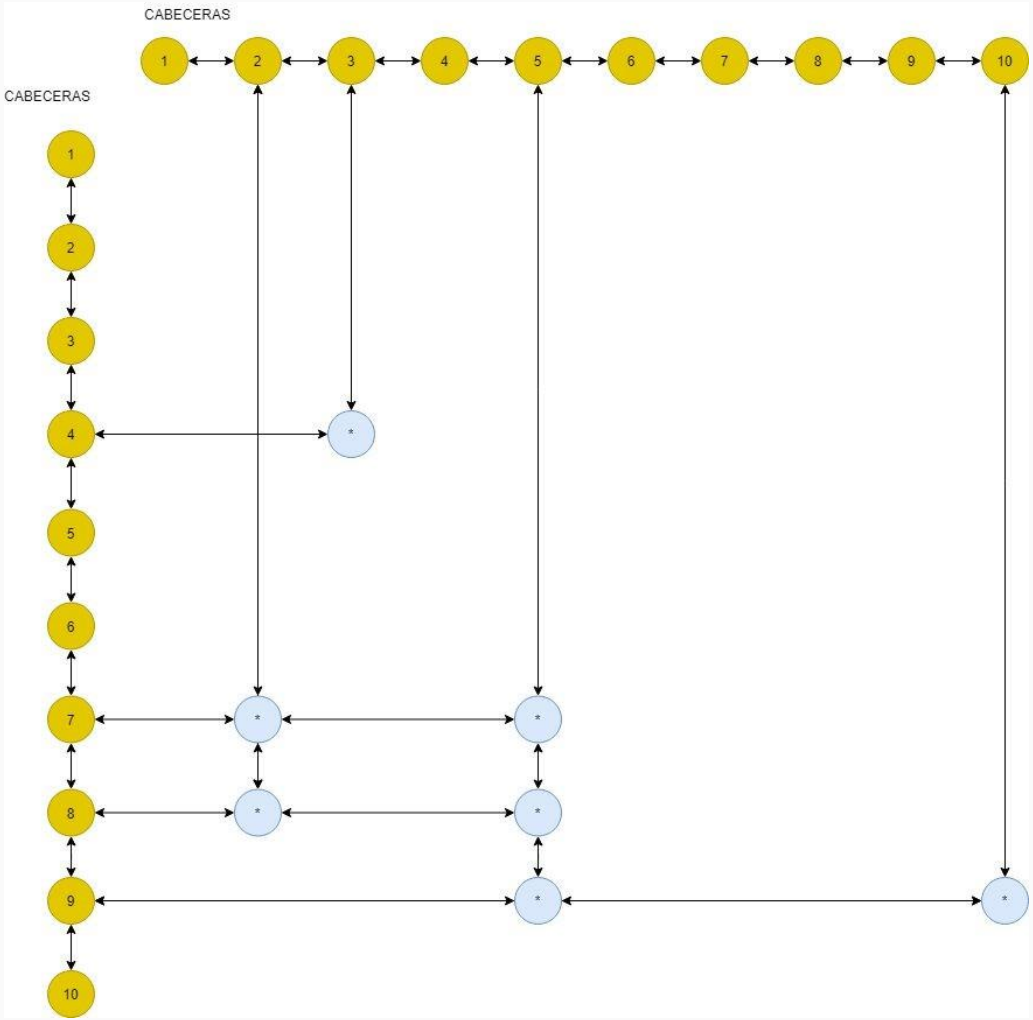


Insertar:

X: 2

Y: 7

Valor: *



Implementaciones

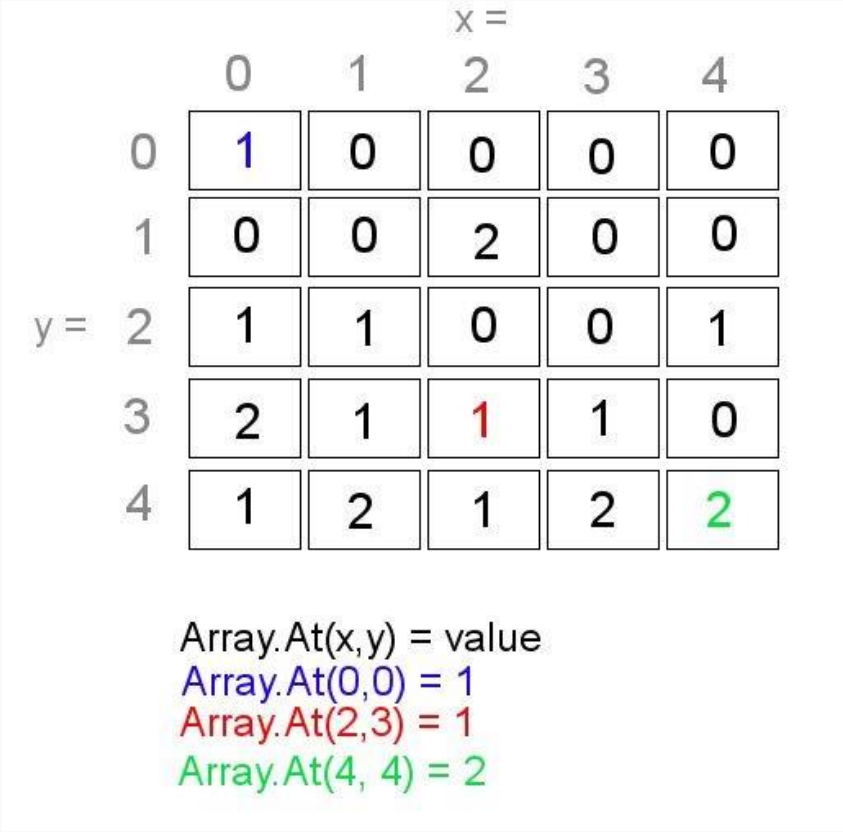
Formas de implementación

Existen diversas formas de implementar una matriz dispersa. Entre las más utilizadas están:

- Arreglos.
  - Existe desperdicio de memoria. Las posiciones 0 (Nulas) existen.
- Lista enlazada.
  - Problemas de rendimiento (inserción, eliminación y búsqueda).
- Enlaces de datos.
  - Mejora el rendimiento, pero consume memoria adicional almacenando los valores de X,Y en los nodos.
- Listas con cabecera.
  - Mejora el rendimiento y reduce el consumo de memoria solo almacenando solo una coordenada en los nodos.

Implementación mediante arreglos

- Generalmente implementado como un arreglo de dos posiciones.
- Es posible utilizar un arreglo de una sola posición, pero es necesario algún método de indexación.
- Muy eficiente para acceder a los datos.
- Existe gran desperdicio de memoria.

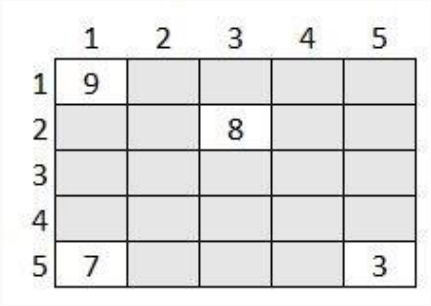


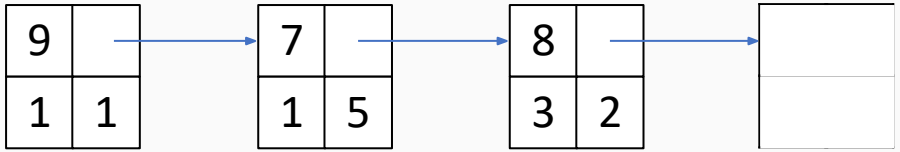
Implementación mediante Lista Enlazada

- Puede ser una lista simple o doble, en la cual los elementos se insertan ordenados.
- Debe de utilizarse algún método de linealización.
- Cada nodo de la estructura guarda su posición (X,Y), su valor y apuntador a siguiente.
- Leve desperdicio de memoria.
- Existen problemas de rendimiento para acceder a los datos.

```
clase Lista{
    clase Nodo{
      int X,Y,valor;
      Nodo sig;
    }

    Nodo primero = nulo;
    ...
    // Operaciones (métodos)
}
```





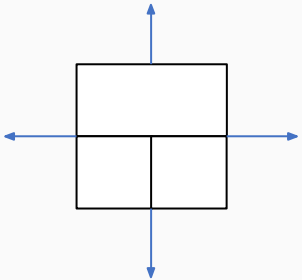
Implementación Mediante Enlaces

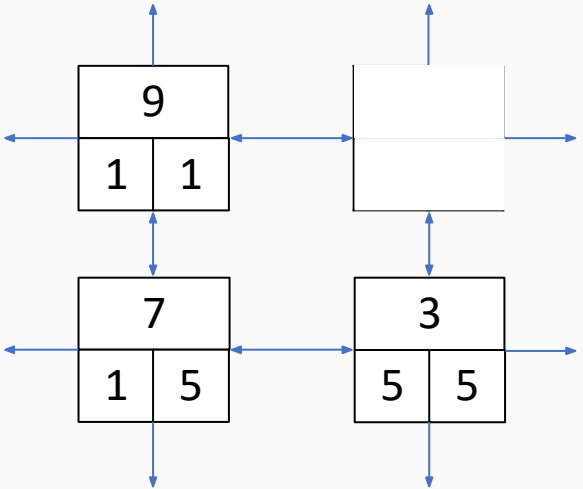
- Puede ser una lista simple o doble, en la cual los elementos se insertan ordenados.
- Cada nodo de la estructura guarda su posición X,Y.
- Cada nodo tiene apuntadores hacia arriba, abajo, siguiente y anterior.
- Existe un leve desperdicio de memoria.
- El redimiendo de las operaciones es optimo.
- Existe una complejidad adicional al enlazar los datos.

```
clase Matriz{
    clase Nodo{
      int X,Y,valor;
      Nodo sig,ant,arriba,abajo;
    }

    Nodo raíz = nulo;
    ...
    // Operaciones (métodos)
}
```

	1	2	3	4	5
1	9				8
2					
3					
4					
5	7				3





Implementación Mediante Lista con Cabecera

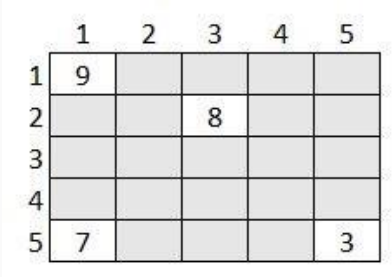
- Puede ser una lista simple o doble, en la cual los elementos se insertan ordenados.
- Cada nodo de la estructura guarda únicamente su valor y los apuntadores necesarios.
- No existe desperdicio de memoria.
- El redimiendo de las operaciones es óptimo.
- Dependiendo de la implementación puede una complejidad adicional al enlazar los datos.

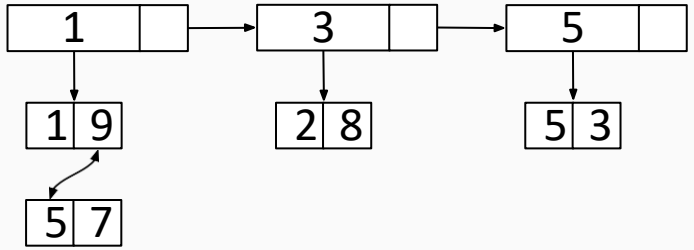
```
clase Matriz{
    clase Nodo{
      Nodo.Celda l;
      Nodo sig;
    }

    Nodo raíz = nulo;
    ...
    // Operaciones (métodos)
}
```

```
clase Lista{
    clase Celda{
      int Y,valor;
      Nodo sig,ant;
    }

    Nodo primero = nulo;
    ...
    // Operaciones (métodos)
}
```





Ejemplo

Matriz ortogonal

Conjunto de listas doblemente enlazadas, accesibles por medio de cabeceras que así como su estructura interna son también listas doblemente enlazadas, unidos entre si por un nodo principal o cabeza principal, que permite el acceso a la estructura