

ESTRUCTURA DE DATOS

AUTOR: WALTER MADRIGAL CHAVES

NOVIEMBRE: 2020



San Marcos

Contenido

INTRODUCCIÓN.....	2
ESTRUCUTURA DE DATOS.....	3
ESTRUCUTURA DE DATOS ESTÁTICAS	3
SIMPLES	4
ESTRUCTURADOS	4
ARREGLOS.....	4
VECTORES.....	5
MATRIZ	6
ARREGLOS MULTIDIMENSIONALES	7
ESTRUCUTURA DE DATOS DINAMICOS.....	7
LISTAS ENLAZADAS.....	8
PILAS.....	10
COLAS.....	12
ARBOLES BINARIOS	13
GRAFOS.....	15
CONCLUSIONES Y RECOMENDACIONES	16
REFERENCIAS BIBLIOGRÁFICAS.....	17



INTRODUCCIÓN

El papel preponderante de la tecnología, el crecimiento de las peticiones en soluciones tecnológicas y la complejidad de los requerimientos, obliga a los desarrolladores de sistemas a estar en constante modernización y cambio. Es necesario ser muy buenos en la creación de herramientas eficientes, que permitan alcanzar el nivel que la demanda del mercado solicita.

El manejo eficiente de los datos dentro de la lógica de un sistema es un reto perpetuo que los programadores constantemente tienen que saltar. La creación de estructuras complejas y dinámicas de datos son alicientes que ayudan a mejorar tanto el rendimiento de software como a mejorar el proceso de su creación.

En la siguiente lectura se analizan estructuras simples y complejas de administración de datos, que se implementa con el fin principal de agilizar y potenciar el desarrollo de grandes sistemas. Algunos de los elementos que abarcan en esta lectura son: Listas enlazadas, Pilas, Colas, Arboles binarios, Grafos, entre otros.

ESTRUCUTURA DE DATOS

La concepción del término "estructura de datos" se enfoca en la forma en que la información se organiza para ser consumida, siempre buscando formas eficientes de almacenamiento y de manipulación. Bajo este concepto los datos se organizan de forma coherente en estructuras especiales, algunas del tipo estáticas y otras dinámicas.

Existen muchos tipos de estructuras de datos, estos se clasifican dependiendo de su complejidad, a continuación, se muestra una imagen en donde se desglosan.

Imagen 1 Tipos de estructura de datos



Fuente: (Nieva Paredes, 2020)

ESTRUCUTURA DE DATOS ESTÁTICAS

Este tipo de estructuras como su nombre lo indican, tienen un espacio fijo o estático en memoria, que es definido desde su concepción y no se modifica su tamaño durante la ejecución del programa. Bajo esta categoría se conocen los

tipos simples y los estructurados.

SIMPLES

Su principal característica es que sólo ocupan un espacio de memoria, la variable bajo este esquema únicamente posee un valor a la vez. Algunos ejemplos son:

- Números enteros.
- Números reales.
- Caracteres.
- Booleanos.

ESTRUCTURADOS

Las estructuras estáticas del tipo estructurados son aquellos que permiten hacer referencia a un grupo de espacios en memoria, permitiendo almacenar un conjunto de datos del mismo tipo. Los arreglos son importante representante de este grupo de estructura de datos.

ARREGLOS

Según (Trejos Buriticá, 2017) Un arreglo es un conjunto de variables en donde cada una de ellas puede ser referenciada utilizando su posición relativa, es decir, su ubicación en relación con el primer elemento de dicho conjunto.

Son estructura de datos estáticas, que administran un conjunto de variables del mismo tipo y bajo un mismo nombre. Hay diferentes tipos, se clasifican según sus dimensiones o extensiones. Los vectores de 1 dimensión, las matrices de 2 dimensiones y por último las de multi dimensiones.

Al momento de crear el arreglo es necesario indicarle elementos importantes como: nombre asignado, tipo de datos que almacenará, cantidad de elementos o tamaño y muy importante cuantas dimensiones tendrá.

Cada integrante de un arreglo es administrado directamente, mediante el nombre del arreglo seguido de una expresión encerrada entre paréntesis, llamada índice del arreglo.

El índice del arreglo es un elemento que permite recorrer toda su estructura, se utiliza para referenciar directamente un elemento del arreglo, toma valores entre uno y la dimensión de este. Si se direcciona una posición que no se encuentra entre dichos límites, se producirá un error de ejecución, pues se estará intentando acceder a una zona de memoria indefinida.

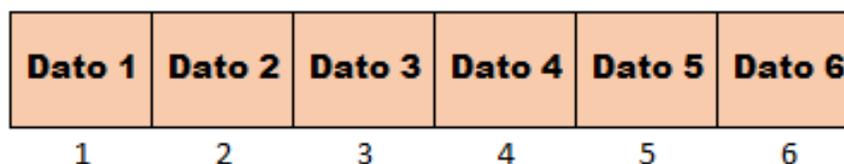
¿Es posible hacer arreglos de clases?

Hay que recordar que un fundamento de la esencia de los arreglos es reunir un conjunto homogéneo de objetos, si se analiza la teoría de la herencia, en la parte que indica que todas las clases tienen un ascendiente común, que es la clase objeto, se puede afirmar que si es posible hacer arreglos de clases.

VECTORES

Son arreglos de una dimensión que permiten almacenar datos sucesivamente de un mismo tipo. La representación gráfica de un vector se muestra en la imagen 2:

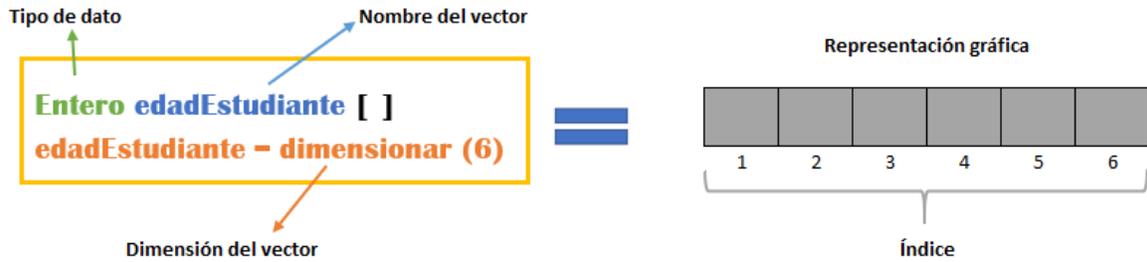
Imagen 2 Representación de un vector



Fuente: Elaboración propia



Imagen 3 forma general para declarar un vector



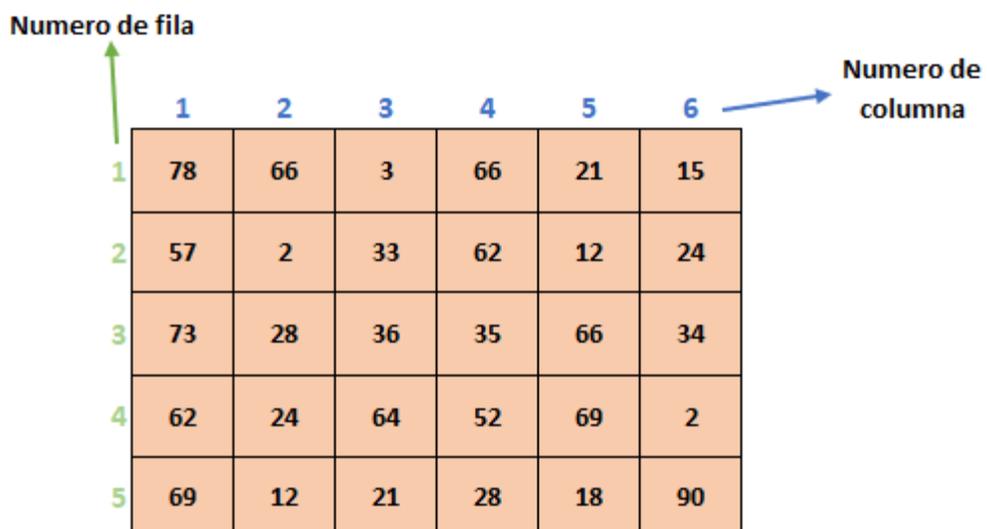
Fuente: Elaboración propia

MATRIZ

Son arreglos de dos dimensiones, al igual que la estructura anterior es indispensable que todos sus datos sean del mismo tipo, su representación gráfica se asemeja a la de una tabla, se compone por una serie de filas y columnas. Se puede decir que es un vector de vectores.

Para recorrer toda su estructura es necesario dos índices, uno que recorra sus columnas y otro las filas.

Imagen 4 Representación gráfica de una matriz



Numero de fila	1	2	3	4	5	6	Numero de columna
1	78	66	3	66	21	15	
2	57	2	33	62	12	24	
3	73	28	36	35	66	34	
4	62	24	64	52	69	2	
5	69	12	21	28	18	90	

Fuente: Elaboración propia

ARREGLOS MULTIDIMENSIONALES

Se definen bajos los mismos principios de las matrices y vectores. La única diferencia es que es una estructura más grande, con más capacidad de almacenar datos y por consiguiente se requieren más índices para recorrer su estructura. Su representación gráfica se asemeja a un cubo.

ESTRUCTURA DE DATOS DINAMICOS

En muchas ocasiones cuando se trabajan con grupos de datos no es posible conocer con anticipación su tamaño, para este tipo de datos es que se crearon las estructuras dinámicas, que como lo indica su nombre, este tipo de estructuras dinámicas aparte de almacenar datos, tienen la capacidad de poder crecer y contraerse según la necesidad real durante la ejecución del programa.

No tienen limitaciones o restricciones en el tamaño de memoria que ocupa. Tienen la capacidad de adaptarse a las necesidades reales a las que suelen enfrentarse los programas. Se clasifican según la forma en que se relacionan sus elementos, en la siguiente tabla se muestran más detalles:

Tabla 1 Tipos de estructura de datos dinámicos

Tipo	Descripción	Ejemplos
Lineal	<ul style="list-style-type: none"> • Los nodos se relacionan de 1 a 1 • Son secuenciales. • Tienen una relación de adyacencia ordenada donde existe un primer 	Pilas- Colas - Listas

	elemento, seguido de un segundo elemento y así sucesivamente hasta llegar al último.	
No lineal	<p>Estructuras de datos jerarquizadas</p> <p>Cada elemento puede estar enlazado a cualquier otro componente</p> <p>Además, cada elemento puede tener varios sucesores y/o varios predecesores.</p>	Árboles, Grafos

Fuente: Elaboración propia

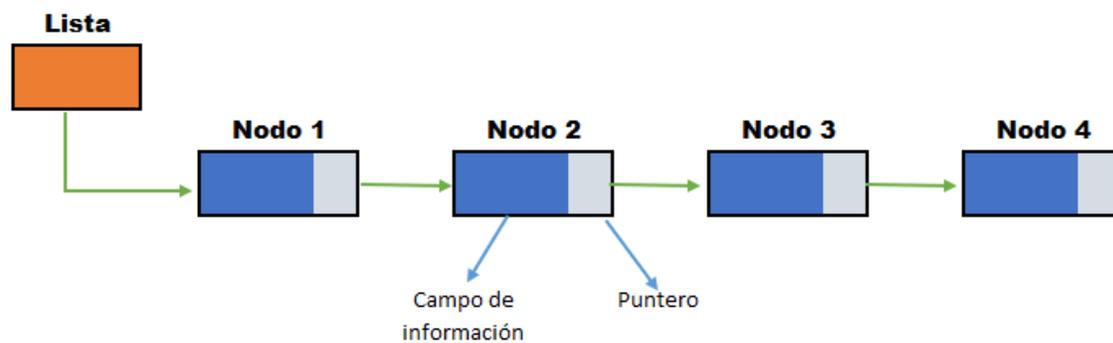
LISTAS ENLAZADAS

La lista es una estructura de datos dinámica lineal, que está compuesta por un conjunto de nodos relacionados, cada uno de ellos posee dos campos: el primero con los datos almacenados (dato simple, estructura de datos, objetos) y el segundo con un apuntador al siguiente nodo de la lista. Además, un apuntador externo señala el primer nodo de la lista.

Una característica de este tipo de estructuras es que durante la ejecución del algoritmo va creciendo o decreciendo según las necesidades que se presenten. Como se indicó en la lista lineal cada elemento apunta al siguiente, es decir, cada elemento tiene información de dónde está el siguiente nodo. Debido a esa estructuración en su forma de accionar recibe el nombre de lista enlazada.

En las listas lineales el primer nodo tiene un papel muy importante, es gracias a el que se puede acceder a toda la lista, por eso es llamado la cabeza de la lista. Eso es porque mediante ese único puntero se puede acceder a toda la lista. Cuando el puntero que se usa para acceder a la lista vale "nulo", se afirma que la lista está vacía.

Tabla 2 Representación gráfica de lista simple



Fuente: Elaboración propia

En la imagen anterior se hace la representación gráfica de una lista, como se observa cada cuadro es un nodo, los cuales se dividen en dos, un espacio mayor para el almacenamiento de los datos y uno más pequeño con la información o apuntador de donde se encuentra el siguiente nodo.

Imagen 5 Listas en Java

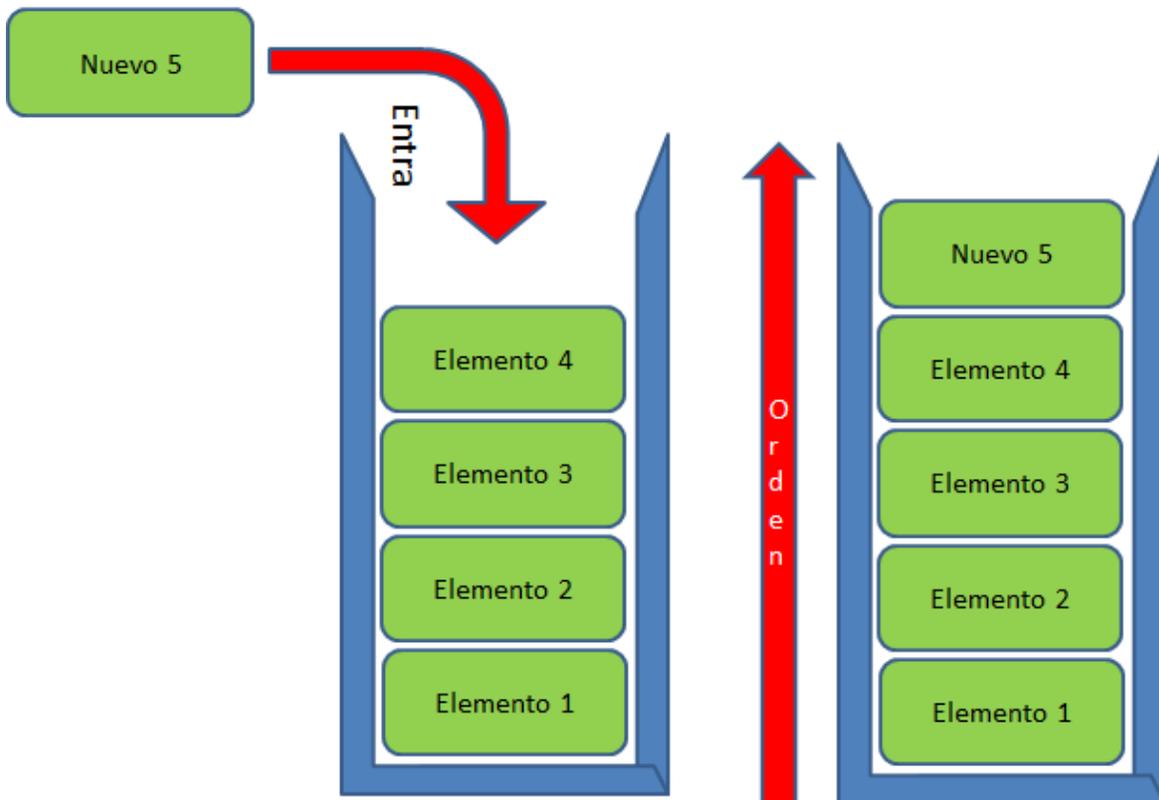
<pre> 1 List list = new ArrayList(); 2 3 4 5 list.add("Andrea"); 6 list.add("Amaya"); 7 list.add("Julio"); 8 9 10 11 Iterator iter = list.iterator(); 12 while (iter.hasNext()) 13 System.out.println(iter.next()); 14 } </pre>	<p>→ La estructura tipo ArrayList es la que se necesita para construir la lista.</p> <p>→ El método add permite agregar datos a la lista directamente.</p> <p>→ La clase Iterator permite obtener el puntero que hace referencia al siguiente elemento de la lista. El método hasNext () indica si hay más elementos y el método. next permite moverse al siguiente elemento.</p>
---	---

Fuente: Elaboración propia

PILAS

Son un tipo especial de lista, también conocidas como listas LIFO que significa: Last In, First Out (El último en entrar es el primero en salir). En este tipo de estructura los elementos se acumulan o apilan, de modo que sólo el elemento que está encima de la pila puede ser leído, y sólo pueden añadirse elementos encima de la pila. Por consiguiente, al agregar un elemento nuevo este quedara de primero para ser leído.

Imagen 6 Representación gráfica de una pila



Fuente: (Blancarte, 2020)

Como se muestra la imagen anterior, las pilas van acumulando datos uno sobre el otro. Lo que hace diferente a otras estructuras es que la pila siempre lee primero el último elemento guardado. Según la imagen el primer elemento en leerse es el #5

Suelen tener 3 operaciones básicas:

- Apilar: añade un elemento a la lista.
- Des apilar: retira un elemento de la lista.
- Verifica: comprueba si la lista está vacía.



Imagen 7 Manipular pila en Java

```

1  Stack pila = new Stack();
2
3
4
5  for (int x=1;x<=10;x++)
6  pila.push(Integer.toString(x
7
8
9
10 while (!pila.empty())
11     System.out.println(pila.pop());
12
    
```

Definir la pila

Insertar elementos a la pila, mediante el método push ()

Vaciado de la pila, el ciclo se ejecuta hasta que la pila este vacía.

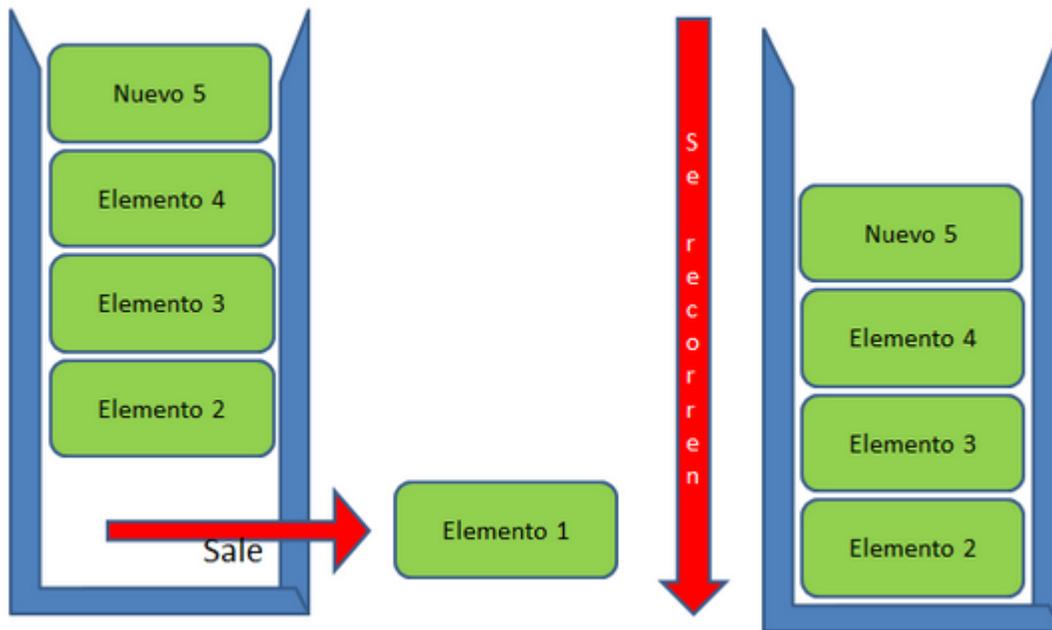
Fuente: Elaboración propia

COLAS

Una Cola es una estructura de datos muy parecida a la pila, la diferencia radica en la forma de leer los elementos, esta estructura utiliza la técnica FIFO: First In-First Out (Primero en entrar primero en salir). Esto quiere decir que el elemento que entre primero a la cola será el primero en leerse.

Esta forma de actuar se presenta en la siguiente imagen, en donde se acumulan una serie de elementos, se denota, que el primero en salir o leerse es el que entro primero a la cola.

Imagen 8 Representación gráfica de una cola



Fuente: (Blancarte, 2020)

Imagen 9 Cola en Java

<pre> 1 Cola cola1=new LinkedList(); 2 3 4 cola.offer(3); 5 cola.add(14); 6 cola.offer(12); 7 8 9 10 while(cola.poll() !=null) 11 System.out.println(cola.peek()); </pre>	<p>→ Definir la cola</p> <p>→ Insertar elementos a la cola, mediante los métodos offer y add</p> <p>→ Vaciado de la cola, el ciclo se ejecuta hasta que la cola este vacía.</p>
---	---

Fuente: Elaboración propia

ARBOLES BINARIOS

Los árboles son estructuras de datos dinámicas, no lineales y bastantes



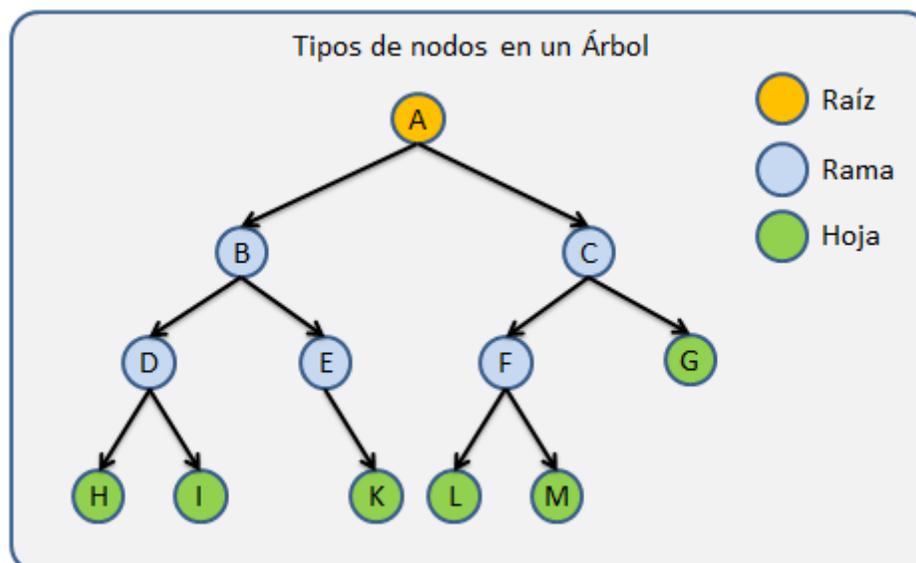
complejas para su implementación. Posee una distribución de nodos jerarquizada y constituida por un conjunto de elementos homogéneos, es decir del mismo tipo.

Esta relación jerárquica entre sus nodos se establece de manera que todos los nodos que componen el árbol tienen un único nodo padre, que es el nodo que le precede en la jerarquía. La única excepción es el nodo raíz, que se puede definir como aquel nodo que no tiene padre. Todos los árboles tienen una única raíz.

Cada nodo puede que tenga muchos hijos o ninguno. Un nodo hijo es el que le sucede en la jerarquía. Si un nodo no tiene ningún hijo se le llama nodo terminal u hoja. A la cantidad de hijos que tiene un nodo se le llama grado.

Los árboles se clasifican según el número de sucesores o descendientes que posee cada nodo, de manera que un árbol donde cada nodo tiene un máximo de dos sucesores se denomina árbol binario, si tiene un máximo de tres sucesores se denomina árbol ternario y así sucesivamente.

Imagen 10 Representación gráfica de un árbol



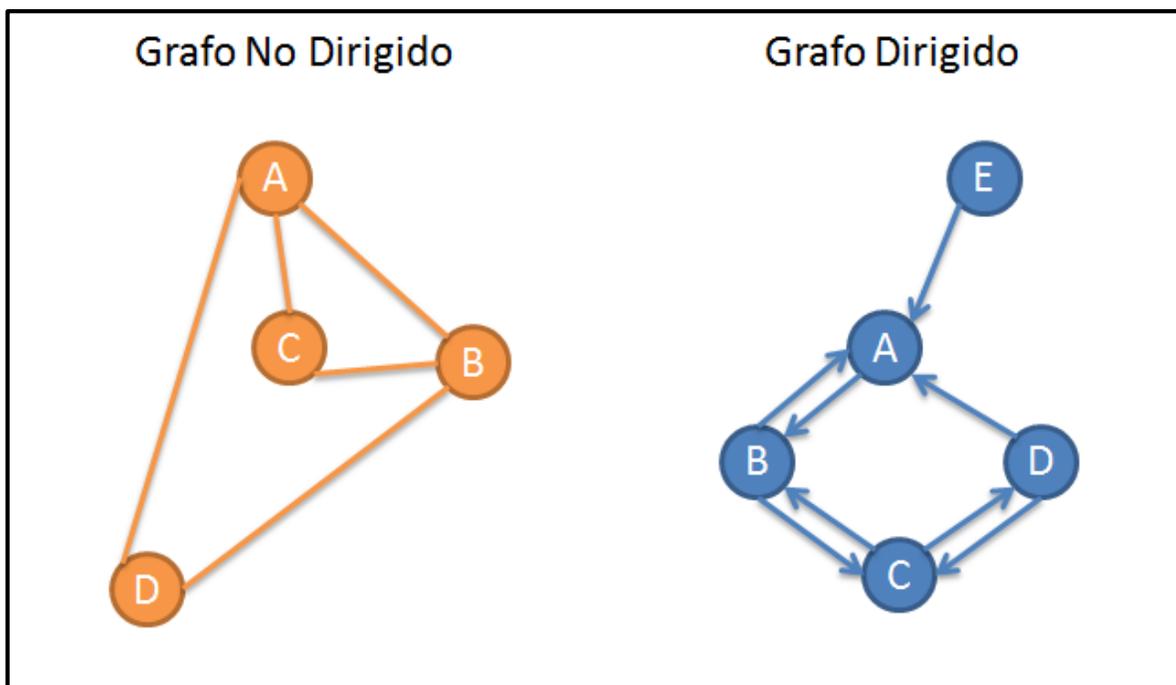
Fuente: (Blancarte, 2020)

GRAFOS

Un grafo es una estructura dinámica y no lineal que es muy parecida a los árboles. Está formado por nodos y otro conjunto de arcos. Un arco es el elemento que agrupan a dos nodos. Cada arco puede tener peso, coste o distancia. Además de arcos pueden estar orientados o no estarlo. En caso de estar orientados se representan mediante una flecha, en caso contrario se representan por un segmento.

Un grafo puede recorrer, agregar, eliminar, modificar elementos igual como lo hace el árbol, bien por profundidad o bien por anchura.

Imagen 11 Representación gráfica de un grafo



Fuente: (Keith, 2020)

Para ahondar más en el tema de Estructura de datos debe realizar la lectura de las páginas 237 a la 272 del libro: Programación orientada a objetos con Java. (2020) de Francisco Blasco.

CONCLUSIONES Y RECOMENDACIONES

- La estructura de datos tiene como objetivo organizar la información de tal manera que sea eficiente su almacenamiento y recuperación. De aquí parte la importancia de dominar procedimientos como el de las pilas, colas, arboles, listas, entre otros
- Es importante recordar que una desventaja de las estructuras estáticas es que su tamaño debe ser definido desde el inicio del programa, impidiendo con esto crecer o disminuir durante la ejecución. Las estructuras dinámicas difieren en este apartado, ellas si son más eficientes en el proceso de su creación, pueden ampliarse o disminuirse según la necesidad que surja.
- Dentro de las estructuras dinámicas existen dos categorías: la lineal y la no lineal. Hay que recordar que la lineal se refiere a cuando los nodos esta ordenados secuencialmente y tienen una relación de adyacencia, mientras que los no lineal son estructuras jerarquizadas en dónde Cada elemento puede estar enlazado a cualquier otro elemento.
- Las pilas y las colas son estructuras de datos que convergen en mucho de su estructura y concepto; difieren en la forma de cómo son tratados los elementos que lo integran. La pila toma y evalúa el elemento más reciente mientras la pila evalúa el elemento más antiguo de ingresar.

REFERENCIAS BIBLIOGRÁFICAS

- Barnes, D., & Kölling, M. (2017). *Programación orientada a objetos con Java usando Bluej 6a. Ed.* Madrid: Pearson.
- Blancarte, O. (15 de 11 de 2020). *oscarblancarteblog*. Obtenido de <https://www.oscarblancarteblog.com/2014/08/06/estructura-de-datos-pila-stack/>
- Blasco, F. (2020). *Programación orientada a objetos en Java*. Bogotá: Ediciones la U.
- Keith, L. (16 de 11 de 2020). *Estructura de datos*. Obtenido de <http://estructurasdedatosk.blogspot.com/2015/12/java-implementacion-de-grafos.html>
- Nieva Paredes, J. (15 de 11 de 2020). *dcodingames*. Obtenido de <https://dcodingames.com/arreglos-unidimensionales-en-java/>
- Trejos Buriticá, O. (2017). *Lógica de programación*. Bogotá: Ediciones de la U.





www.usanmarcos.ac.cr

San José, Costa Rica