



# ESTRUCTURA DE DATOS

ING. JULIA ANDREA PINEDA ACERO

UNIVERSIDAD SAN BUENAVENTURA CALI

(BEL, 2020)

# TDA (TIPO DE DATOS ABSTRACTOS)

Las estructuras de datos están presentes en la vida cotidiana de las personas de manera implícita, de hecho es común interactuar con estas diariamente. Por ejemplo, cuando se dobla la ropa se la suele apilar, en el supermercado se debe hacer cola para poder pagar, cuando se realiza una actividad muchas veces se hace una lista de tareas que conforman la misma, etcétera. Al momento de representar el modelo real de estas estructuras en un modelo virtual es necesario extraer de estas sus principales características, que permitan representarla virtualmente, y su mecánica de funcionamiento.

# TDA (TIPO DE DATOS ABSTRACTOS)

El TDA permite definir atributos que describen las características deseadas del modelo real y funciones que representan las actividades que se pueden realizar con el mismo. Con esta técnica se logra una representación parcial del modelo real, suficiente para poder representar dicho modelo y usarlo de manera virtual.

Existen dos alternativas para realizar estos TDA, utilizando variables estáticas —mediante el uso de vectores o matrices— o variables dinámicas —recurriendo al uso de punteros y variables dinámicas—. Cabe destacar que dependiendo del modelo a representar con una simple estructura estática es más que suficiente y no es necesario recurrir a estructuras dinámicas.

TDA, para representar las distintas estructuras de datos clásicas de las ciencias de la computación:

- Pila
- Cola
- Lista
- Tabla de dispersión
- Árbol
- Montículo
- Grafo

# CONCEPTOS PREVIOS

Variable dinámica: aquella que se puede crear y destruir en tiempo de ejecución, a diferencia de una variable estática, que lo hace en tiempo de compilación.

Este tipo de variables son administradas mediante el uso de punteros.

Puntero: el puntero es un tipo de dato que se utiliza para almacenar una dirección de memoria de una variable dinámica —es decir se podría decir que un puntero apunta a una variable dinámica—.

Se conoce también como dirección de referencia.

# ESTÁTICO VS. DINÁMICO

```
n1 = 10  
n2 = n1  
n2 = n2+15  
print("Valor n1:", n1, "- Valor n2:", n2)
```

```
l1 = [5, 10, 15]  
l2 = l1  
l2[1] = 50  
l2.append(150)  
print("Valor l1:", l1, "- Valor l2:", l2)
```

# PUNTERO

```
p1=[0,1]
p2=[0,1]

print (id(p1))
print (id(p2))

if(p1 is p2):
    print("p1 y p2 apuntan a la misma variable dinámica")
else:
    print("p1 y p2 apuntan a diferente variable dinámica")

p1={}
p2=p1

print (id(p1))
print (id(p2))

if(p1 is p2):
    print("p1 y p2 apuntan a la misma variable dinámica")
else:
    print("p1 y p2 apuntan a diferente variable dinámica")
```



# PUNTERO

El único valor conocido que el programador le puede asignar a un puntero es vacío o None (en el caso de Python) —dado que no es posible conocer las direcciones de memoria—. Esto significa que el puntero no apunta a ninguna variable, también se puede asignar una dirección de memoria que esté almacenada en otro puntero. Las demás direcciones son asignadas automáticamente por el sistema operativo cuando se crean las variables dinámicas.



# CONCEPTOS PREVIOS

Variables huérfanas: cuando una variable dinámica deja de ser apuntada por un puntero, quedará en memoria pero no podrá accederse a ella porque su dirección se perdió.

En Python cuando ocurre esto el intérprete elimina automáticamente dicha variable para liberar espacio en memoria. Por lo que para eliminar una variable dinámica solo basta con cambiar la dirección del puntero que la señala.

# CONCEPTOS PREVIOS

Nodo: es un tipo de datos que se define mediante el uso de una clase —dado que en Python no existen los registros— para representar a cada elemento de nuestra estructura de datos.

Dependiendo de esta pueden variar sus atributos, pero en su esencia un nodo debe tener mínimamente un campo donde se almacene la información y otro que lo enlace con el siguiente o anterior nodo de la estructura.

# CONCEPTOS PREVIOS

Un nodo simplemente enlazado básicamente está compuesto por dos campos, información (o dato) y siguiente (o enlace).




El campo información puede ser un tipo de dato simple o una clase que agrupe un conjunto de atributos y el campo siguiente es un puntero que almacenará la dirección del siguiente o anterior elemento de la estructura (nodo).



# NODO

```
class Nodo(object):  
    info, sig = None, None  
  
aux=Nodo()  
aux.info="Primer nodo"  
palabra=input('Ingrese una palabra: ')  
naux=aux  
while (palabra != ""):  
    nodo=Nodo()  
    nodo.info=palabra  
    naux.sig=nodo  
    naux=nodo  
    palabra = input ('Ingrese una palabra: ')  
  
while (aux is not None):  
    print(aux.info)  
    aux=aux.sig
```



Un TDA está compuesto de dos bloques básicos, estructura o definición de tipos e interfaz o comportamiento.

En el primero de estos bloques se hace la definición formal de la estructura es decir las características que se utilizan para representar el modelo real de manera virtual.

Mientras que en el segundo se define mediante funciones el comportamiento de dicho modelo; estas funciones serán los eventos mediante los cuales se debe manejar su funcionamiento.

