

# Estructuras de Datos

Ayudantía 3



Unidad Básica

# Nodos

- Unidad básica de una estructura de datos.
- Posee:
  - Un **valor**, información que queremos almacenar.
  - **Referencias** a otros nodos.

# Nodos

>>> Ejemplo

```
class Nodo:
```

```
    def __init__(self, value=None, otro_nodo=None):
```

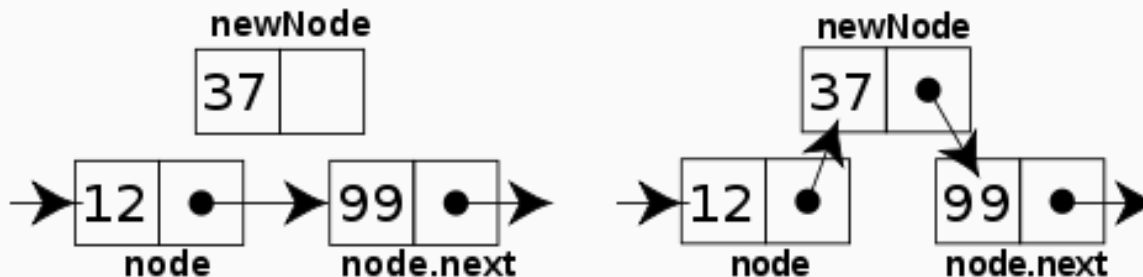
```
        self.value = value
```

```
        self.otro_nodo = otro_nodo
```

# Listas Ligadas

# Listas ligadas

Es una colección lineal de elementos, en que cada uno tiene referencia al **siguiente** elemento.



Estas siempre tienen una cabeza y una cola. Al agregar un nodo, el nodo de la cola agrega una referencia al nodo nuevo.

# Listas ligadas

```
class ListaLigada:
    def __init__(self):
        self.head = Nodo()
        self.length = 1 #opcional

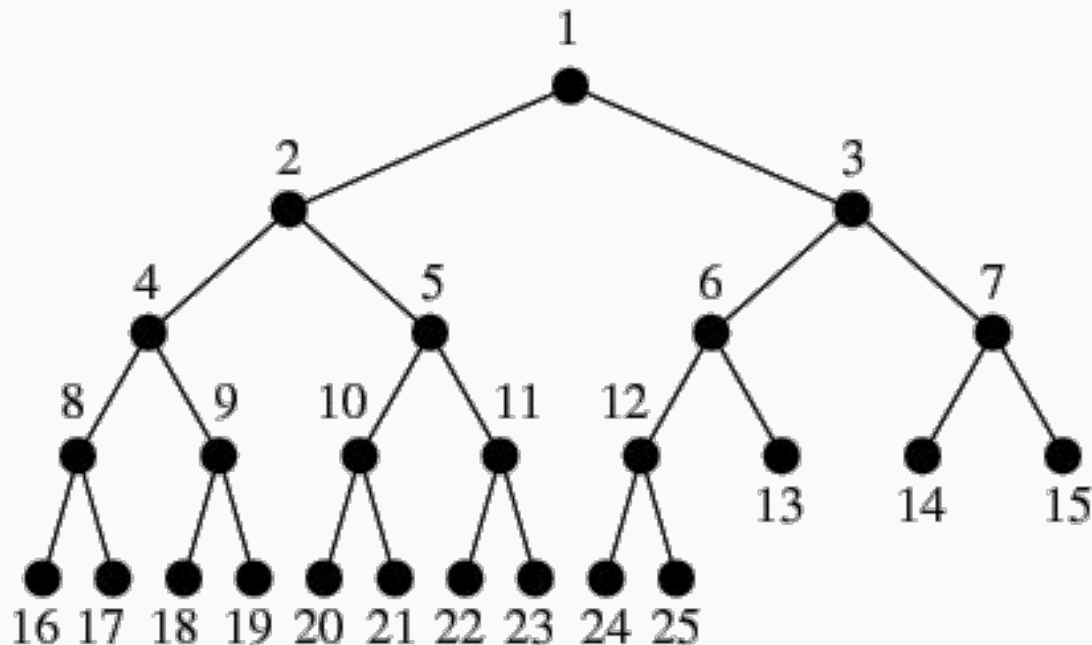
    def get_posicion(self, n):
        nodo = self.head
        for i in range(n):
            if nodo.next == None:
                return None
            nodo = nodo.next
        return nodo
```

# Árboles



# Árboles binarios

Son estructuras de nodos en que cada nodo tiene 0, 1 o 2 hijos y solo un padre. Hay un nodo raíz que es el primero en ver en el árbol.



# Árboles binarios

Ejemplo:

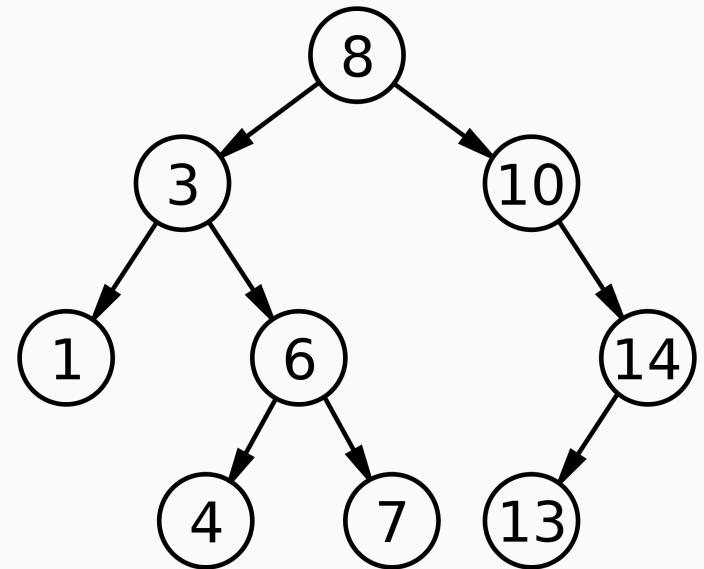
```
class Nodo():  
  
    def __init__(self, value=None, left=None, right=None, dad=None)  
        # Referencia al padre depende de la funcionalidad del árbol  
  
        self.value = value  
        self.left = left  
        self.right = right  
        #self.dad = dad
```

# Árboles de búsqueda binarios

Son un caso particular de árboles binarios.

Un elemento a la izquierda de un nodo es menor que el nodo actual.

Un elemento a la derecha de un nodo es mayor que el nodo actual.



# Árboles de búsqueda binarios

```
class Nodo_ABB():
    def insertNode(self, otro_nodo):
        nodo_actual = self
        if otro_nodo < nodo_actual:
            if nodo_actual.left == None:
                nodo_actual.left = otro_nodo
            else:
                nodo_actual.left.insertNode(otro_nodo)
        # Es lo mismo para el lado derecho?
```

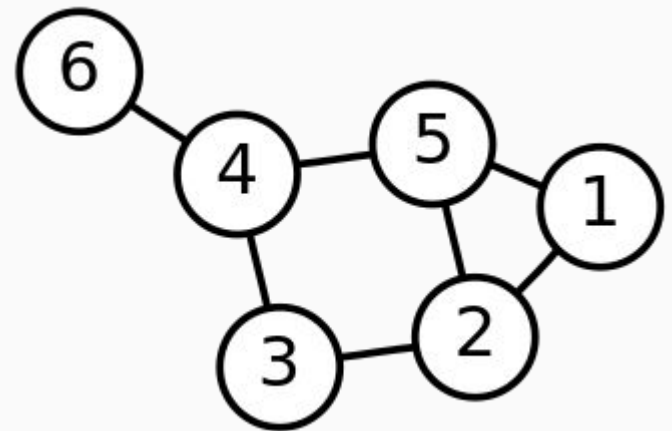
Grafos

# Grafos

Son un conjunto de nodos y arcos, que producen relaciones entre si.

Cada nodo puede estar relacionado con uno o más nodos.

Pueden ser dirigidos o no dirigidos.



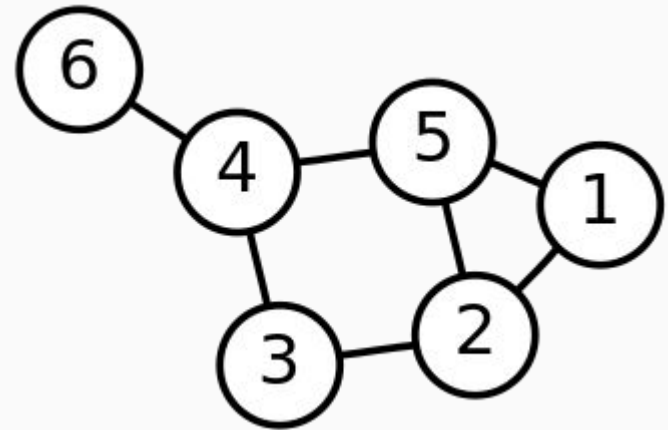
# Grafos

Matemáticamente se definen como  $G = (V, E)$ , donde  $V$  son los vértices y  $E$  el conjunto de tuplas que relacionan cada nodo con otro.

Ejemplo imagen anterior:

$$V = \{1, 2, 3, 4, 5, 6\}$$

$$E = \{(1, 2), (2, 3), (3, 4), (4, 5), (5, 1), (4, 6)\}$$



# Grafos

```
class Nodo():  
    def __init__(self, value=None, vecinos=None):  
        self.value = value  
        if vecinos is None:  
            vecinos = list()  
        self.vecinos = vecinos  
  
    def agregar_nodo(self, otro_nodo):  
        self.vecinos.append(otro_nodo)  
        otro_nodo.vecinos.append(self)
```



	<i>Lista ligada</i>	<i>Árbol binario</i>	<i>Grafos</i>
<i>Referencia al padre</i>	Solo si es doblemente ligada.	Opcional	Solo si no es dirigido
<i>Número de referencias a otros nodos</i>	1	2	oo