



Lista simplemente enlazada

Son una estructura de datos fundamental en la programación, se utiliza para almacenar elementos de manera secuencial, almacena una lista de elementos y la manera en que podemos movernos es mediante un puntero.

Su funcionamiento es mediante una colección de nodos y cada nodo contiene la data y un apuntador hacia el siguiente nodo:



FIG 00: representación de un nodo al lado de su respectivo código.

Para poder utilizar esto necesitamos crear algo que se llama lista “simplemente enlazada”

Una lista simplemente enlazada es la colección de nodos que nos va a permitir guardar la información, dicha clase contiene un apuntador y varios datos... si observa la siguiente figura usted podrá notar cómo funciona:

Lista simplemente enlazada

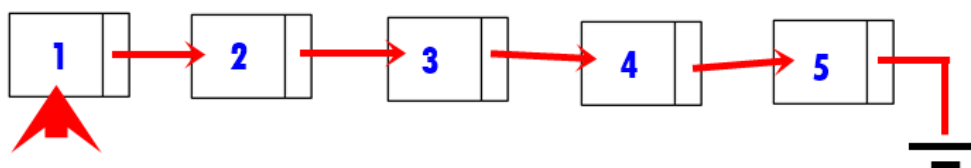


FIG 01: Ejemplo visual de una lista simplemente enlazada.

Las listas necesitan los siguientes atributos y métodos para poder funcionar:

- Se necesita un pivote el cual es el encargado de moverse para leer/escribir la información.
- Se necesita un método para agregar.
- Se necesita un método para mostrar toda la lista.

Advertencia: con esos pocos métodos nuestra lista es funcional.

Que es el pivote

El pivote no es más que una instancia del nodo, pero no podemos empezar la lista instanciando un nodo sin una data (o al menos no es recomendable) el pivote es la cabeza del nodo y se encarga de moverse para recorrer la lista y poder agregar elementos.

El pivote/cabecal es el encargado de ser el punto de acceso a los datos de la lista sin él no existiría una forma de recorrer la lista.

```
class SinglyList:
    def __init__(self) -> None:
        self.pivot = None
```

Advertencia: el pivote inicia en None dado a que cuando la lista es creada no posee datos.

Como se agregan elementos a la lista

Yo he decidido hacerlo de manera recursiva aunque también es posible hacerlo de manera iterativa. para agregar a la lista se tiene que partir de 2 casos base:

- Caso 1: la lista está vacía: en ese caso la solución es muy simple entonces el pivote se declara como nodo y se le ingresa el dato.

```
def addData(self, data):
    if self.pivot == None:
        self.pivot = Node(data)
```

- Caso 2: La lista no está vacía en ese caso lo que hacemos es crear un nuevo nodo donde va a estar contenida el nuevo dato y luego recursivamente empezamos a recorrer los nodos de siguiente en siguiente hasta encontrar el siguiente que se encuentre vacío y guardamos el nuevo nodo en la siguiente posición vacía.

```
def _addData(self, pivot, new_node):
    if pivot != None:
        if pivot.next == None:
            pivot.next = new_node
        else:
            self._addData(pivot.next, new_node)
```

Métodos que le dan valor agregado a la lista:

- Se necesita un método para contar el total de elementos de la lista.
- Se necesita un método para verificar si existe un dato en la lista.
- Se necesita un método para actualizar un valor de la lista.
- Se necesita un método para eliminar un elemento.
- Se necesita un método para saber si la lista está vacía.
- Se necesita un método para obtener un dato en una posición x.