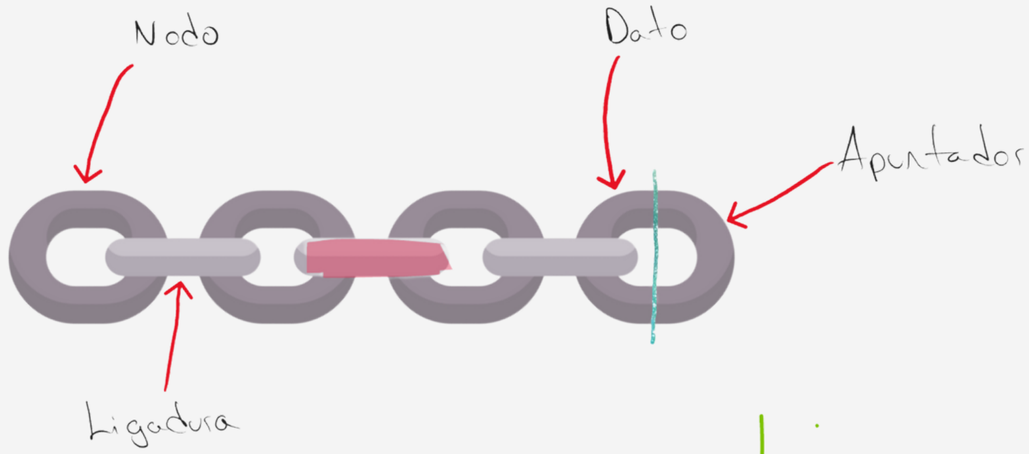
3001A-V22 ESTRUCTURA DE DATOS Y ALGORITMOS

LISTAS LIGADAS IMPLEMENTACIÓN

Profesor: Rafael Pérez Aguirre

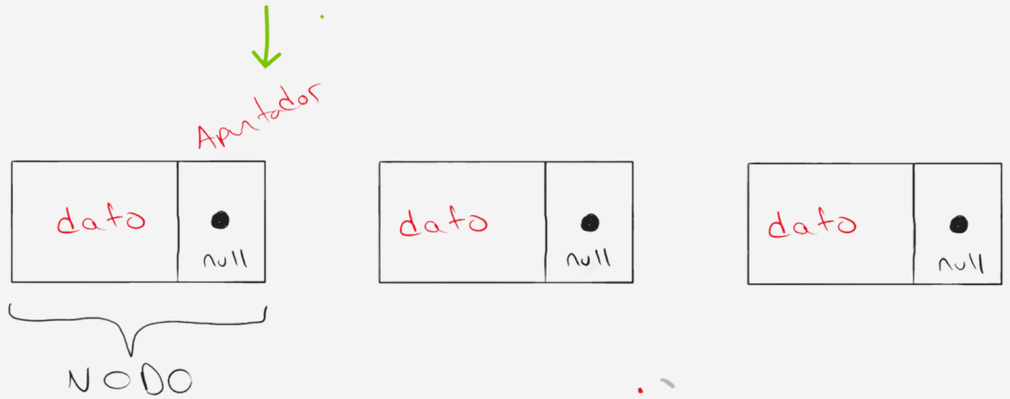
VERANO 2022

Análisis



Analisis

- Comprobar si la cabeza esta vacía
- Si la cabeza esta vacía, el nodo será la cabeza
- Si la cabeza no esta vacía, obtener el ultimo nodo
- Asignar el nuevo nodo al puntero del siguiente



Métodos

Listas ligadas

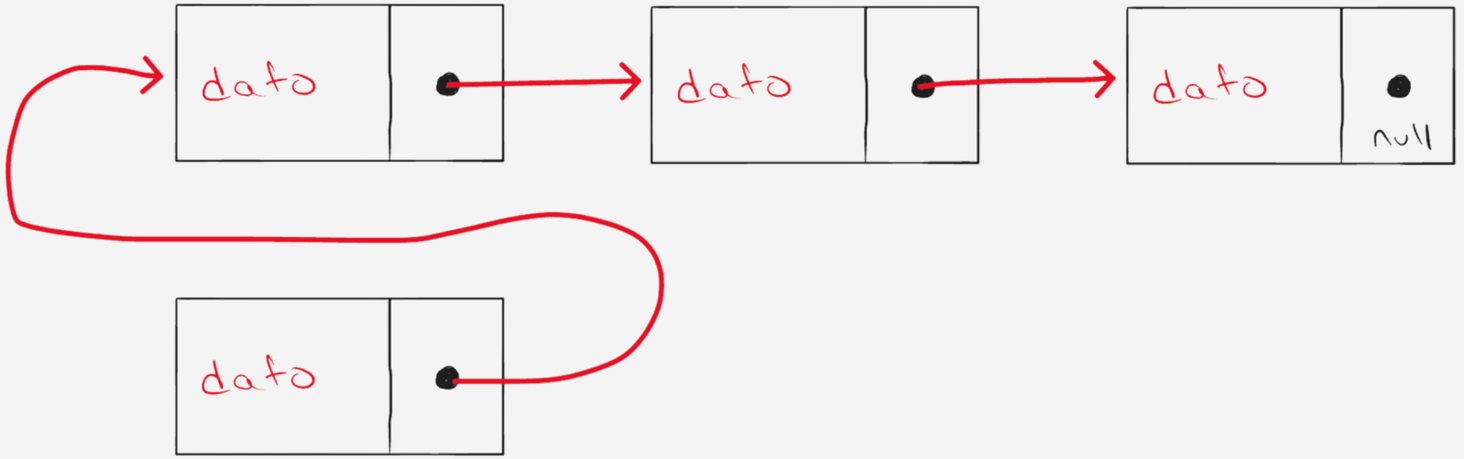
Métodos de inserción

- Último
- Inicio
- Posición

null

null null

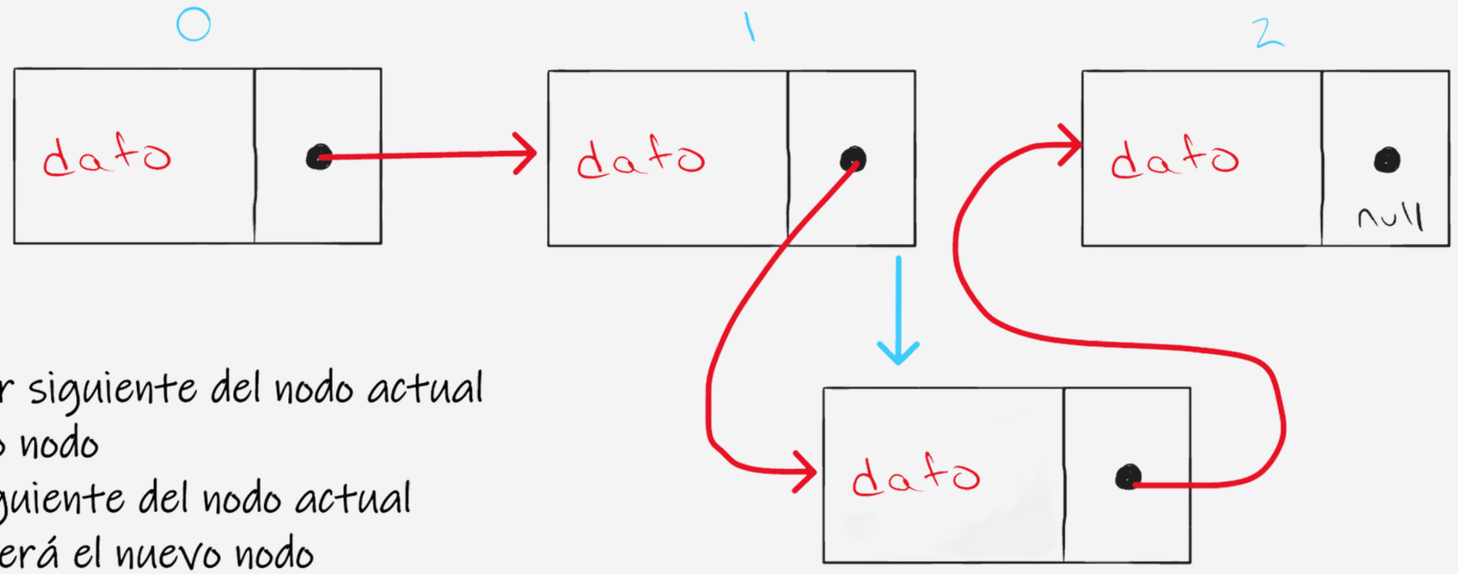
Insertar al inicio



Inserción al inicio

Métodos

Insertar en una posición



1. Copiar siguiente del nodo actual al nuevo nodo
2. El siguiente del nodo actual ahora será el nuevo nodo

Inserción en una posición

The code

lista_ligada.py

```python
class Nodo:
    def __init__(self, dato):
        self.dato = dato
        self.siguiente = None

class ListaLigada:
    def __init__(self):
        self.cabeza = None
        self.contador = 0

    def insertar_final(self, nuevo_nodo):
        # Comprobar si la lista ligada esta vacía
        if self.cabeza:
            # Obtener el ultimo nodo
            ultimo_nodo = self.cabeza
            while ultimo_nodo.siguiente != None:
                ultimo_nodo = ultimo_nodo.siguiente
            # Asignar el nuevo nodo al siguiente puntero del ultimo nodo
            ultimo_nodo.siguiente = nuevo_nodo
        else: #La cabeza esta vacía
            # Asignamos el nodo a la cabeza
            self.cabeza = nuevo_nodo
        self.contador = self.contador + 1

    def insertar_inicio(self, nuevo_nodo):
        primer_nodo = self.cabeza
        self.cabeza = nuevo_nodo
        nuevo_nodo.siguiente = primer_nodo
        self.contador = self.contador + 1

    def insertar_pos(self, nuevo_nodo, pos):
        if self.cabeza:
            if pos == 0:
                self.insertar_inicio(nuevo_nodo)
            elif pos >= self.contador:
                print('Ese indice no existe, overflow!')
            else:
                print('Buscando...')
                i = 0
                actual_nodo = self.cabeza
                while i < pos and actual_nodo.siguiente != None:
                    print(f'Buscando... {i}')
                    actual_nodo = actual_nodo.siguiente
                    i += 1
                print(f'Te encontre! pos: {i}, dato: {actual_nodo.dato}')
                nuevo_nodo.siguiente = actual_nodo.siguiente
                actual_nodo.siguiente = nuevo_nodo
        else:
            print('Aún no tengo datos :(')
        self.contador = self.contador + 1

    def cantidad(self):
        return self.contador

    def mostrar(self):
        nodo_temporal = self.cabeza
        while nodo_temporal != None:
            print(f'{nodo_temporal.dato}', end='->')
            nodo_temporal = nodo_temporal.siguiente
        print('Null')

    def obtener(self, pos):
        if self.cabeza:
            if pos == 0:
                nodo = self.cabeza
                return nodo.dato
            elif pos >= self.contador:
                return None
            else:
                i = 0
                actual_nodo = self.cabeza
                while i < pos and actual_nodo.siguiente != None:
                    actual_nodo = actual_nodo.siguiente
                    i += 1
                return actual_nodo.dato
        else:
            return None #Aun no tengo datos

    # TAREA
    def eliminar_inicio(self):
        pass

    def eliminar_pos(self, pos):
        pass
```